UNIVERSITÉ DU LUXEMBOURG

PhD-FSTC-2016-6
The Faculty of Sciences, Technology and Communication

DISSERTATION

Defense held on 24/02/2016 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

Sune S. NIELSEN

BORN ON 15 JULY 1977 IN FREDERIKSSUND, DENMARK

DIVERSITY PRESERVING GENETIC ALGORITHMS - APPLICATION TO THE INVERTED FOLDING PROBLEM AND ANALOGOUS FORMULATED BENCHMARKS

Dissertation defense committee

Dr. Reinhard Schneider, Chairman
Head of Bioinformatics Core Facility, LCSB, University of Luxembourg

Prof. Dr. El-Ghazali Talbi, Vice Chairman
Professor, LIFL, Université de Lille

Prof. Dr. Pascal Bouvry, Dissertation Supervisor
Professor, CSC, University of Luxembourg

Dr. Grégoire Danoy, Member
Research Assistant, CSC, University of Luxembourg

Dr. Wiktor Jurkowski, Member
Group Leader, The Genome Analytics Center, Norwich, UK

Dr. Roland Krause, External Expert
Research Assistant, LCSB, University of Luxembourg

Dedicated to Lucy, Mick and Romy.

Abstract

Protein structure prediction is an essential step in understanding the molecular mechanisms of living cells with widespread applications in biotechnology and health. Among the open problems in the field, the Inverse Folding Problem (IFP) that consists in finding sequences that fold into a defined structure is, in itself, an important research problem at the heart of most rational protein design approaches. In brief, solutions to the IFP are protein sequences that will fold into a given protein structure, contrary to conventional structure prediction where the solution consists of the structure into which a given sequence folds. This inverse approach is viewed as a simplification due to the fact that the near infinite number of structure conformations of a protein can be disregarded, and only sequence to structure compatibility needs to be determined. Additional emphasis has been put on the generation of many sequences dissimilar from the known reference sequence instead of finding only one solution. To solve the IFP computationally, a novel formulation of the problem was proposed in which possible problem solutions are evaluated in terms of their predicted secondary structure match. In addition, two specialised Genetic Algorithms (GAs) were developed specifically for solving the IFP problem and compared with existing algorithms in terms of performance. Experimental results outlined the superior performance of the developed algorithms, both in terms of model score and diversity of the generated sets of problem solutions, i.e. new protein sequences. A number of landscape analysis experiments were conducted on the IFP model, enabling the development of an original benchmark suite of analogous problems. These benchmarks were shown to share many characteristics with their IFP model counterparts, but are executable in a fraction of the time. To validate the IFP model and the algorithm output, a subset of the generated solutions were selected for further inspection through full tertiary structure prediction and comparison to the original protein structure. Congruence was then assessed by super-positioning and secondary structure annotation statistics. The results demonstrated that an optimisation process relying on a fast secondary structure approximation, such as the IFP model, permits to obtain meaningful sequences.

Contents

Acknowledgements

- My deepest gratitude goes to Prof. Pascal Bouvry for making this thesis possible and for the valuable sessions and indispensable expert advice and direction.
- A special word of thanks goes to Dr. Grégoire Danoy for the many hours spent exchanging ideas and for collaboration in research.
- Many thanks also go to Dr. Reinhard Schneider as well as current and former members of his team at the LCSB, Dr. Roland Krause and Dr. Wiktor Jurkowski, for providing valuable background and support in biology and protein research.
- Thanks to Christof Ferreira Torres for help on last minute experiments and to the rest of team Bouvry for an unmatched great social work environment.
- Finally I would like to thank friends and family for their support and understanding these last years.
- Work funded by the National Research Fund of Luxembourg (FNR) as part of the EVOPERF project at the University of Luxembourg with the AFR contract no. 1356145. Experiments were carried out using the HPC facility of the University of Luxembourg [66]

1

Introduction

1.1 Context

Proteins are large biomolecules organised in long chains of amino acids or residues. Each joint in this chain is flexible, but due to intermolecular forces they rarely exist as flexible chains, but fold up into more compact structures. Better packed structures are also more stable and will degenerate slower in nature than poorly folded proteins do. This in turn increases the chances of survival and also explains why naturally occurring proteins are mostly neatly folded. A simplified illustration of a real protein is provided in Fig. 1.1. Proteins are also responsible for molecular functions in the cells found in all living organisms, be it micro-organisms, plants or animals, but also exist and function outside of cells. They mainly catalyse (i.e. help or support) chemical reactions, but can also act as inhibitors that prevent certain reactions from happening. Finally they can have mechanical functions such as building blocks in muscles or cell structure. Especially in the latter examples, but also in general, the structure is very important for the function of the protein. Hence, the relation between the amino acid sequence of a protein and its three-dimensional structure is one key research topic of structural biology. Obtaining the folded structure of an arbitrary protein sequence allows functional studies by computer simulation rather than by expensive experiments in the wet lab and has given rise to the field of protein engineering.

Protein engineering in general aims at designing molecules with desired properties and a method that allows to successfully design such molecules would find applications in a huge number of fields from health to biotechnology applications. For example, it could allow to design improved enzymes for biotechnology applications such as waste-water treatment or biomass production [11]. For medical purposes, new antibodies can be designed specifically towards already known targets. For example, a given pathogen like HIV, could be targeted by an engineered protein designed for binding to its envelope spikes and ultimately neutralising the virus [37].

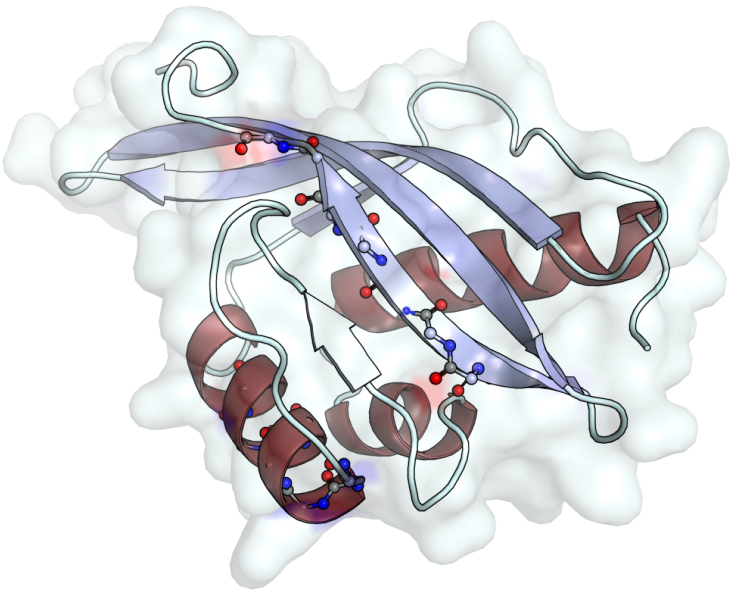


Figure 1.1: Protein example *1OH0* with its surface shown semi-transparent. *Helix* and *sheet* secondary structure segments are shown in dark red and light blue respectively. Selected atoms of the molecule are displayed for further clarification.

1.2 Motivation

Since the advent of genome sequencing, all protein-coding genes of an organism can be obtained with ease but protein structure prediction capabilities were only slightly improved over the last two decades. Most approaches in protein engineering depart from a known structure and replace only a limited number of amino acids in the sequence, or make use of large databases of protein fragments. If however, no homologous structure to a given sequence exists (i.e. the *ab-initio* problem), finding the correct structure essentially remains an intractable task. This hampers even the comparably easy task of classifying protein sequences into families. The *ab-initio* protein folding problem is unsolved due to a number of reasons, most prominently due to the enormous complexity of the search space of possible structure states. In addition, the actual simulation of forces depends on an a multitude of factors and interactions which have not been fully understood and perfectly modelled by researchers. These include both forces acting internally among the atoms of the molecule, but also interactions with the surrounding chemical environment where temperature, charge and acidity level plays a role. Finally protein folding cannot be solved accurately as a static problem, but must be understood as a process with many intermediate folded states, referred to as the *folding pathway*, before reaching the final form. In contrast

to the *ab-initio* protein folding problem discussed so far stands the Inverse Folding Problem (IFP). The IFP is an early suggested method [18] of simplifying the seemingly impossible task of full *ab-initio* structure prediction. The idea is to depart from the known wild type structure (i.e., the structure of the original protein as it occurs in nature) and predict sequences that would match the known fold, rather than finding the fold for a given sequence. While still very difficult, this approach may be slightly less complex, especially when starting from the known wild type sequence and only exchanging a subset of the amino acids. As it turns out, this can be quite useful for protein engineering, where a specific structure is sought after. Generally there will be some limitations or preferences regarding the choice of sequence, but an ultimate goal is to achieve maximum freedom with respect to the protein sequence to design both structure and surface properties. Another research question arising in this context is that of determining *how many* new sequences can be found which will lead to the same structure as the wild type. At the same time researchers seek new sequences as different as possible from the known wild type and its homologs (i.e., related proteins sharing traits from common distant ancestors). To respond to these requirements and propose answers to the posed questions, methods are required to efficiently explore the large solution space consisting of all possible protein sequences. As a result of such an exploration of the sequence space is expected a collection of diversified solutions.

1.3 Contributions

The primary goal of this thesis is to find many diversified solutions to the presented Inverse Folding Problem (IFP) for proteins. A model based on secondary structure prediction of real sequences was developed for use with evolutionary algorithms. Based on this work, an analogue benchmark was designed to exhibit many of the same characteristics allowing to efficiently test and compare different algorithms and algorithm settings. Two algorithms were designed specifically for the task of optimising the developed models yielding diversified collections of prospect solutions to the IFP. With the right settings the algorithms show superior performance in terms of fitness and diversity at the same time when compared to existing algorithms. A large collection of generated sequences with good model scores had their structure predicted in CPU-intensive simulations amounting to a total of almost two years of CPU time. By comparing to their respective reference structure, up to one in five of generated proteins were predicted to have the same fold as their target.
Below are listed the main contributions:

- Model for secondary structure evaluation with respect to a reference protein molecule
- A benchmark model mimicking many characteristics of the real-world problem computed in a fraction of the time

- A Diversity as Objective (DAO) evolutionary algorithm with tunable diversity emphasis by means of the added Quantile Constraint (QC) method
- A Preference Based Genetic Algorithm (PBGA) with an even more direct adjustment of diversity emphasis and huge potential for further development by modifying the preference equation
- Large scale automated validations of found solutions to the real-world problem by means of a state-of-the-art tertiary structure prediction method.

1.4 Outline

This dissertation is divided into 4 parts. Part I covers two main subjects in this thesis: Protein Engineering and Optimisation. First the Inverted Folding Problem for proteins is presented with related work in protein design. Next optimisation heuristics background and related work in diversity preserving is discussed. In Part II the optimisation model developed is presented along with analysis and definition of an analogous benchmark model. Then two new algorithms and their respective mechanisms and settings which have been developed as part of this thesis are described. Part III presents experimental setup and experiments conducted on real-world and benchmark problems. Finally the dissertation is concluded in Part IV.

Part I

Related Work and Problem Description

2

Problem Description

2.1 Introduction

In the following, the subject of protein folding with related definitions and minimal background is introduced in Section 2.2. Then the specific type of inverse protein folding is described and the stated requirements which constitute the target problem to be tackled by this work are detailed in Section 2.3 and 2.4. Discussing only the problem background, and not yet it's solution, emphasis here is on the field of biology.

2.2 Background Notions

As discussed, proteins are large biomolecules organised in long chains of amino acids or residues. The following section will give fundamental background knowledge of amino-acids and protein structures to arm the reader for the sections on protein folding.

2.2.1 Amino Acids

Amino Acids are organic building blocks consisting mainly of Carbon (C), Hydrogen (H), Oxygen (O) and Nitrogen (N) atoms. Common for all amino acids are their *amine* and *carboxylic acid* functional groups. There are 20 different abundantly occurring amino acids in nature, see Table 2.1, which vary in size and electrical characteristics. Technically, many more are possible, some of which are also found in living organisms, but the details are beyond the scope of this work. Figure 2.1 shows three of the 20 amino-acids, also referred to as residues, in chemical notation together with their three-dimensional molecular structure. Oxygen atoms are coloured red, Nitrogen blue, Carbon light blue and Hydrogen grey.

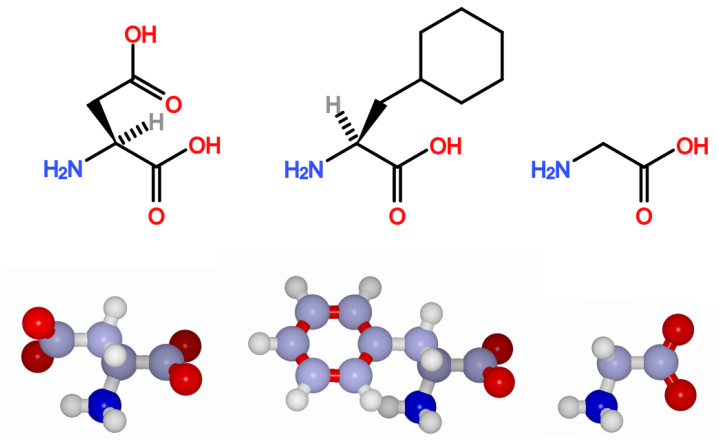


Figure 2.1: Three amino acids with chemical symbol and stick model of their atoms. From left to right: Aspartic Acid (Asp), Phenylalanine (Phe) and Glycine (Gly)

Table 2.1: The 20 amino-acids

Name	**Abr.**	**Letter**	**DNA Codons**
Alanine	Ala	A	GCT, GCC, GCA, GCG
Arginine	Arg	R	CGT, CGC, CGA, CGG, AGA, AGG
Asparagine	Asn	N	AAT, AAC
Aspartic acid	Asp	D	GAT, GAC
Cysteine	Cys	C	TGT, TGC
Glutamic acid	Glu	E	GAA, GAG
Glutamine	Gln	Q	CAA, CAG
Glycine	Gly	G	GGT, GGC, GGA, GGG
Histidine	His	H	CAT, CAC
Isoleucine	Ile	I	ATT, ATC, ATA
Leucine	Leu	L	CTT, CTC, CTA, CTG, TTA, TTG
Lysine	Lys	K	AAA, AAG
Methionine	Met	M	ATG
Phenylalanine	Phe	F	TTT, TTC
Proline	Pro	P	CCT, CCC, CCA, CCG
Serine	Ser	S	TCT, TCC, TCA, TCG, AGT, AGC
Threonine	Thr	T	ACT, ACC, ACA, ACG
Tryptophan	Trp	W	TGG
Tyrosine	Tyr	Y	TAT, TAC
Valine	Val	V	GTT, GTC, GTA, GTG
Stop codons	Stop		TAA, TAG, TGA

2.2.2 Protein Structure

The structure of a protein can be described on three levels, which will each be discussed in further detail below:

- Primary structure - the protein sequence
- Secondary structure - the notation of structure elements
- Tertiary structure - the three-dimensional organisation of all atoms

The protein sequence is the code that describes the linear combination of any of the common amino-acids which bind through peptide bonds to form the protein backbone of $N - C_\alpha - C$ atoms as shown at the bottom of Figure 2.2. When ordered

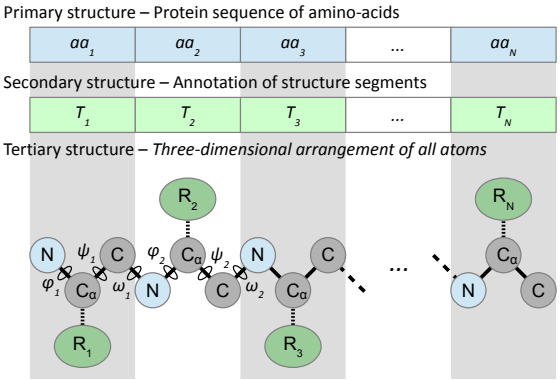


Figure 2.2: Three levels of protein structure

from left to right, as in the figure, the *amine* group, here represented by its Nitrogen (N) atom for simplicity, is situated to the left of the amino acid, respectively at the beginning of the chain.

The side-chains, noted as R_i, vary with each of the possible amino acids, both in terms of size and other properties, such as charge, acidity and hydropathy. A typical protein sequence is 50 to 300 residues long. Due to the rotational freedom of the atom bonds and the molecular forces acting between the residues it folds into one canonical three-dimensional structure. These intermolecular forces are the sum of a number of complex interaction forces largely depending on the mentioned properties of the residues, but also on the distance and orientation of interacting atoms and structures. In general the protein structure will try to adapt a lower energy configuration like a bolder will roll down a mountain into the valley due to the gravitational force. In the case of proteins, such a more relaxed state corresponds to parts of the protein being either stacked or curled together referred to as *sheets* or *helices* as seen in Fig. 1.1. The remaining unstructured segments are commonly referred to as *loops* and serve as flexible connections between the other segments. The structure of a protein can be defined in different levels (see Fig. 2.2). The primary structure is the protein sequence

of N amino acids $\{aa_i\}$ where $1 \leq i \leq N$ is the residue position. The secondary structure defines the organisation of *helices*, *sheets* and *loops* of the tertiary structure and can be expressed by a type $\{T_i\} \in \{H, E, L\}$ for each position i in the protein. If for example a protein consists of a *helix* and two *sheets*, its secondary structure could be described by such a list: $\{L, L, H, H, H, H, L, E, E, L, E, E, L\}$ - though this example is significantly shortened for simplicity. The tertiary structure completely describes the arrangement of all atoms of a protein in the three-dimensional space. There is a definite relation between secondary and tertiary structure levels. The tertiary structure imposes the secondary structure, but a given secondary structure may produce several different tertiary structures. As the arrangement and size of structural elements are dictated by the secondary structure, limiting the amount of different well folded tertiary structures obtainable with the same secondary structure. The ensemble of three-dimensional positions of C_α atoms is commonly referred to as the alpha-trace which provides a rough residue type- and rotation-independent view of the proteins configuration. Similar protein sequences generally obtain the same configuration or fold but sequences not recognisable by similarity can nevertheless fold into 3D-structures that are easily brought into congruence.

2.3 Protein Folding

Protein folding and protein prediction research is concerned with finding or predicting the folded structure of a given amino acid sequence. Obtaining the sequence of existing proteins is relatively easy compared to obtaining the structure of these through computer simulation. Simulations are only partially successful and the only way to validate is to synthesise a designed protein sequence in a laboratory and establish the structure (e.g.: through Nuclear magnetic resonance (NMR) spectroscopy, cryo-electron microscopy or x-ray crystallography). All of this is very time consuming and expensive especially when a large batch of sequences are to be probed. Many research teams have contributed to the field over the last three decades developing a number of theories and software frameworks.In an effort to unify the efforts, and provide a fair foundation for comparison, the CASP competition, or experiment as it is also referred to, was first held in 1994 [49]. Since then it has taken place every two years with the latest published resume in [48] of the $10th$ competition held in 2012. CASP stands for "Critical Assessment of protein Structure Prediction" and researchers in the experiment receive sequences of proteins with unpublished structures and submit their own predicted structures for evaluation. Though the CASP experiments have had many successes, the progress through the last decade seems to have come to a hold with no reliably overall well performing methods and little new ideas.

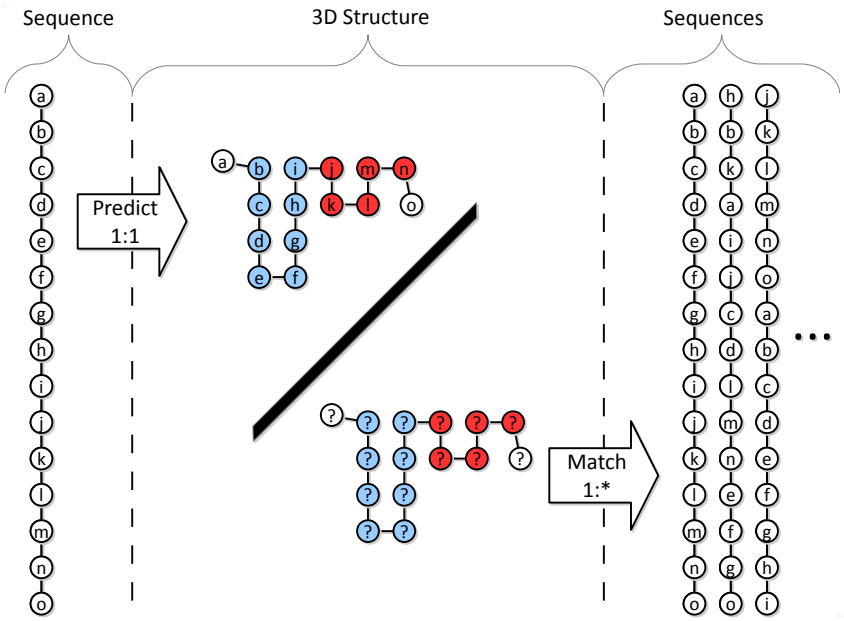


Figure 2.3: Conventional fold prediction vs. inverse fold matching. Sheet and helix positions on the structure are marked blue and red respectively.

2.4 Inverse Protein Folding

As the problem of protein folding is not solved, scientists have early on sought to simplify the task by solving the inverse problem. With the hierarchical definition of Fig. 2.2 in mind, the Inverse Folding Problem (IFP) can be defined as follows: given a primary structure (protein sequences) and its corresponding tertiary structure, find alternative sequences that will result in the same tertiary structure. Fig. 2.3 shows a schematic simplification of conventional and inverse folding. The inverted problem is thought of as a simplification because the structure is given, and sequence to structure compatibility becomes the main difficulty, i.e. determining what combination of amino acids to fit the '?' positions in the figure. When the structure is unknown (the *ab-initio* case) the number of possible configuration solutions is enormous. A central part of any protein design-process is to obtain, or come close to, a target tertiary structure with a certain degree of freedom in the choice of protein sequence. Hence solving the IFP would be a key to successfully engineer proteins. Furthermore, the IFP is of general scientific interest to study the size, shape and characteristics of the sequence space that matches a given target structure. Clearly such studies would benefit from access to large collections of sequences all related to the structure under investigation. In this work, the fact that matching secondary structures is a necessary, but not a sufficient condition for proteins to have the same tertiary structures, is exploited to reduce the IFP to its simplest formulation: given a protein's secondary structure and

its corresponding protein sequence as input, find a set of highly dis-similar protein sequences that will result in the most similar secondary structure. Diversity among the found sequences is hence a requirement as the goal is not to find one sequence, but many new sequences representing a larger area of the feasible solutions space.

2.5 Conclusions

After a brief introduction to proteins, their structure and folding, the challenging Inverse Folding Problem (IFP) has been presented. Opposite to protein folding, a solution to the IFP consists in generating a *sequence* that matches the given reference structure. The significance of solving the IFP for real-world successful protein design has likewise been established. Judging by the advancements in protein folding, it seems that alternative approaches are welcome to shed light on new aspects of the folding problem, but also to enable further studies of the sequences to structure relationship. The approach to the IFP presented in this chapter distinguishes itself in two aspects. First of all, the returned result shall not be a single matching sequence, but a large amount of diverse sequences. Secondly, to gain an advantage in computation cost, the evaluation of sequences is to be based solely on secondary structure prediction which is not a guarantee, but a requirement to achieve the target tertiary structure. The formulated IFP instance seems to be a suitable target problem for a population-based metaheuristic search method due to the large amount of diverse sequences to be returned and the enormous search space arising from 50 to 300 residue long sequences.

3

Genetic Algorithms and Optimisation Techniques

3.1 Introduction

Optimisation is a field with a long history that originates in mathematics. It covers numerous sub fields and techniques and can find application in any situation where designs or decisions need to be optimised with respect to given criteria. A prerequisite though is that the optimiser can receive feedback of some sort on how well these criteria are met for a proposed design or decision set. One large sub category of optimisation methods are inspired by evolutionary concepts found in nature with the goal of adapting to a difficult problem - that of efficient survival of the species. These nature-inspired approximate methods are generally more suited for problems of high complexity where exhaustive sampling is not feasible. This is the case when finding the optimal solution cannot be guaranteed in polynomial time, given that the problem instance is large enough. For this reason, evolutionary algorithms cannot guarantee to find the best possible solution in limited time, but rather seek to find near optimal solutions to the problem at hand. Under the common classification of metaheuristic optimisation algorithms, this work focuses on a sub class: that of genetic algorithms. These consist in evolving one or more populations of individuals with a genetic coding representation that corresponds to solutions to the problem at hand. In the rest of this chapter basic history and background is discussed in Section 3.2 with general and example optimisation problems presented in Sections 3.3 and 3.4. The standard generational genetic algorithm and genetic operators as well as the multi-objective extension is detailed in Section 3.5.

3.2 Optimisation and Search Heuristics

The field of optimisation techniques can be divided into two main sub classifications consisting of *exact* and *approximate* methods, see Fig. 3.1. Exact techniques con-

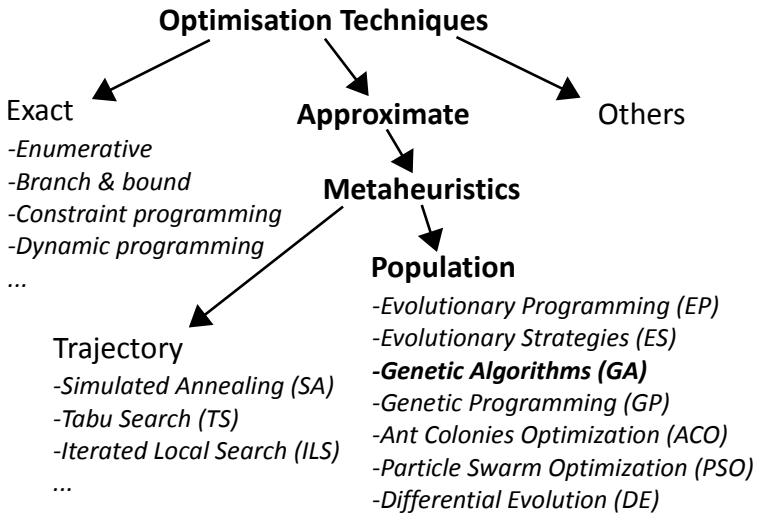


Figure 3.1: Optimisation methods classification

sist of iterative and enumerate methods, Branch & Bound approaches among many others. The latter is a widely used method continuously dividing the search space and progressing through a search-tree. Branches of the tree determined to consist of an inferior quality solution space are pruned and disregarded by the search process. Under approximate methods the large scientific field of metaheuristics is found with a large portion of population based algorithms. The basic evolutionary ideas as presented so far originate from before computers were capable of actually running them. Since the sixties variations of the idea were developed independently by different scientists: Evolutionary Programming (EP) [21], Evolutionary Strategies (ES) [56, 57] and Genetic Algorithms (GA) [29] which all fall under the common term of Evolutionary Algorithms (EA). The GA is characterised by employing a fixed length chromosome also referred to as *genotype* to represent candidate problem solutions. These are evolved with selection, crossover and mutation without exploiting specific problem knowledge. In contrast, ES use problem-dependent solution representation and generally make no use of crossover, but rather adaptive step sizes. EP is closely related to ES, and also use problem dependent representations and solution perturbations (i.e. mutations). Where ES normally uses deterministic selection eliminating poor individuals, EP will use stochastic selection and tournament type mechanisms. For a more detailed overview of the field of metaheuristics [63] is an excellent read.

3.3 General Optimisation Problem

Before applying an optimisation algorithm, a problem model and solution representation must first be designed. This representation generally consists of a vector $\mathbf{x} = \{x_i\}$ of i decision variables each being a real or a discrete value.

To evaluate the fitness of a problem solution in the decision (or solution) space Ω, $\Omega \subset \mathbb{R}^n \vee \Omega \subset \mathbb{N}^n$, one takes its corresponding representation $\mathbf{x} \in \Omega$ and passes it to the problem optimisation model. This model is defined by the objective function f in Equation 3.1 and constraint functions g_j in Equation 3.2. The model evaluates a solution and returns a fitness in the objective space of real values $\mathbb{R}$ and a constraint penalty in case constraints are not respected:

$$f : \mathbf{x} \in \Omega \rightarrow \mathbb{R}, \tag{3.1}$$

In addition to the fitness of an individual an arbitrary number of constraint functions g_j may be defined to identify feasible solutions and limit the size of the valid search space Ω:

$$\begin{aligned} &g_j : \mathbf{x} \in \Omega \rightarrow \mathbb{R}, \\ &g_j(\mathbf{x}) \geq 0 \end{aligned} \tag{3.2}$$

3.3.1 Common Challenges and Goals in Optimisation

It may seem obvious, but the overall goal is always to obtain one or more optimal solutions to the problem at hand. Typically the *global optimum* is sought after, which is *the* individual solution to the problem with the best fitness. Depending on the problem it may not be trivial at all, especially if many local optima exist, i.e. solutions which have the best fitness in the local neighbourhood of the solutions space. Such problems are often referred to as *rugged*, *multi-modal* and sometimes *deceptive* as discussed in Section 5.2. The risk of getting stuck in a local optima and finding ways of escaping them might be one of the most important issues in metaheuristics literature.

3.4 Sample Optimisation Problem

To illustrate the concepts, a simple problem example is presented: a light technician needs to place a number of spotlights with the goal of covering a theatre stage the best possible. The solution vector $\mathbf{x}$ would then define the placement and orientation of all available spotlights while the model would be able to simulate the result in terms of illumination of the stage and the set.

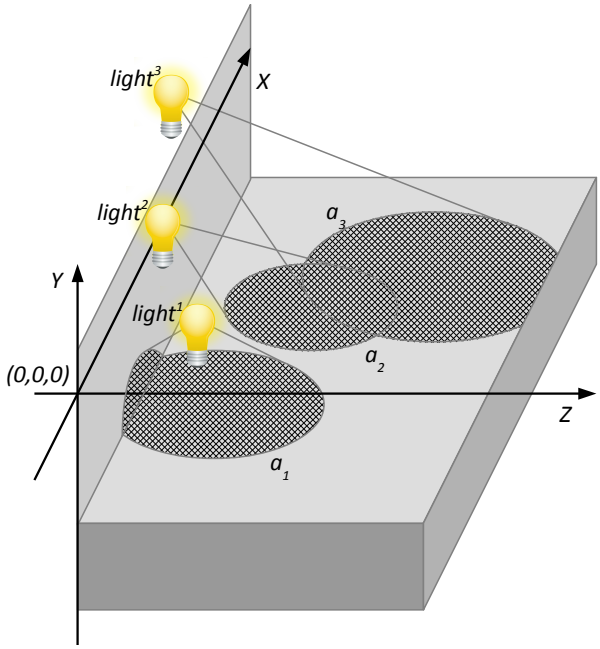


Figure 3.2: Illumination of a stage as an optimisation problem.

3.4.1 Objective Function

For the sake of simplicity, lets define the objective function f_{area} as the union of stage surface area a_i covered by each of the N spots:

$$f_{total}(\mathbf{x}) = \bigcup_{i=1}^{N} a_i \tag{3.3}$$

The goal would be to maximise the function of Eq. 3.3 to produce a light configuration covering the largest possible stage surface area. How the decision vector (*genotype*) is translated into an actual light configuration(*phenotype*) depends on the solution encoding. A Straightforward approach would be to systematically store all six degrees of freedom for all N lights as:

$$\mathbf{x} = \{light^1_{posX}, light^1_{posY}, light^1_{posZ}, light^1_{rotX}, light^1_{rotY}, ..., light^N_{rotZ}\}$$

This example leads to a very long solution vector and can in addition represent redundant solutions. For example if the six degrees of freedom are swapped between two spotlights, they technically swap places, but the resulting lightning would be the same. Or if you add 360 degrees to any rotation it leads to the same orientation of the spotlight.

3.4.2 Constraints and Encoding

Many solutions will be illegal, for example those representing physically overlapping spotlights or positions below the floor or above the ceiling. Other solutions can immediately be discarded, for example if any ceiling mounted lights point at the ceiling or into the crowd. This is where valid variable ranges and constraints come into the picture. Variable minimum and maximum values are usually defined per variable, but it may not be enough and actual constraints are needed to ensure that valid solutions are produced. For example one constraint could be that of placing the lights ordered from 1 to N from left to right:

$$light^1_{posX} < light^2_{posX} < light^N_{posX}$$

Another constraint could limit lights from pointing upward if placed near the ceiling, and downward if placed on the floor. Say $posY$ defines the height above ground with *zero* set to $1m$ above ground.

$$light^i_{posY} < 0 \Rightarrow light^i_{rotX} > 0,\ light^i_{posY} > 0 \Rightarrow light^i_{rotX} < 0$$

In practice there will be a limited number of positions available on few mounting rails and the spots orientation can only be adjusted in steps around two axes. If such limitations are recognized by the designer, the encoding can be greatly simplified. Instead of having $3xN$ real position variables, each light could be assigned one of a limited number of possible positions resulting in only N discrete position variables and so on.

3.5 Evolutionary and Genetic Algorithms

For an evolutionary process to proceed, a *genotype* coding representation for individuals and an optimisation model of the problem needs to be defined, see Section 3.3 and 3.4. The *genotype* of an individual can be passed to the optimisation model returning a fitness value for the individual. Now selection, recombination and mutation can be simulated to preserve and generate fitter individuals throughout the execution of the algorithm - emulating the well-known *survival of the fittest* principle for species in their natural environment. After a number of generations the algorithm will terminate based on a stopping criterion, and the final population of individuals represent good solutions to the optimisation problem at hand. Usual stopping criteria are *time*, *number of evaluations*, *number of generations* or convergence towards an acceptable good fitness value among others. In this thesis the main focus is on GAs which were first introduced by Holland [29]. The approach of general evolutionary algorithms is independent of the underlying optimisation problem, hence requires no knowledge of the nature of the problem which to the algorithm is essentially a *black box*. Though an infinite amount of algorithm designs and tuning variations can be thought of, some will work better for one class of problems but worse for others and vice versa. In fact no algorithm will outperform all others on all problems - a theorem

referred to as the "No free lunch" theorem [73], hence choices need to be made by the algorithm designer or user based on the nature of the problem to be solved.

3.5.1 Generational Genetic Algorithm and Genetic Operators

With the ability to evaluate and represent individual solutions as a vector coding $\mathbf{x}$, the Genetic Algorithm (GA) essentially simulates a simple natural environment where individuals mutate and are selected for mating and survival. Three basic operators are essential to the algorithm, namely *selection*, *mutation* and *crossover*:

$$\begin{aligned} f_s &: \Omega^N \to \Omega \\ f_m &: \Omega \to \Omega \\ f_c &: \Omega^2 \to \Omega^2 \end{aligned} \tag{3.4}$$

The selection operator f_s of Equation 3.4 will select an individual from a set of individuals. A widely used *selection* operator is the *tournament*, or *binary tournament operator* which returns the better of two randomly drawn individuals, hence the term “binary”. The operator is called twice to select two parents for mating by the *crossover* operator.

With the stage lightning model and the objective function in Section 3.4 in mind, the concept and power of the *crossover* operator can be explained. Obviously a solution where all spots illuminate the same point would result in a relatively small area covered, hence be a bad solution. On the other hand two individually poor solutions that are good in covering only the front and the rear part of the scene respectively may yield a better solution covering the entire scene if properly combined. The recombination can be achieved by the evolutionary *crossover* operator by mating two parent solutions as seen in Figure 3.3. A solution consists in the configuration of four lights, here simplified with *one* solution loci assigned to describing the orientation of *one* light, giving a solution length of four.

The *mutation* operator takes as input a solution genotype, and perturbs or mutates it. Typically, the mutation probability p_m is set to $1/N$ and a draw is done for each N loci in the solution vector. In this case it means that on average, one loci per individual will be mutated.

Using these operators the algorithm proceeds by iteratively selecting, recombining and mutating individuals as shown in Algorithm 1. The algorithm starts by initialising a population X_0 with random generated individuals before evolving them one generation at the time until the available time t_{max} (here equivalent to the number of generations) has passed. During one generation, $N/2$ pairs of parents are selected for mating producing two children each. In this basic version the offspring population of children completely replace their parents.

3.5.2 Exploration vs Exploitation

A crucial characteristic of any algorithm is the trade-of between exploration and exploitation. It is thought of as a balance because in order to explore many widespread

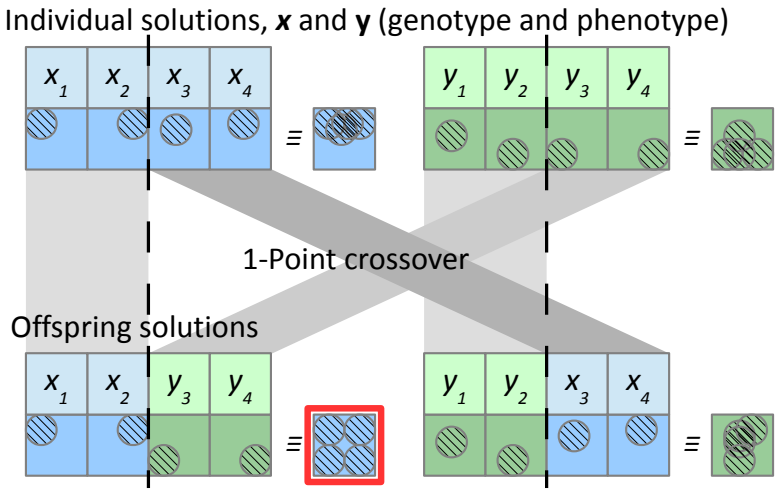


Figure 3.3: Crossover on two solutions to the stage lightning example.

areas of the search space, one cannot allocate as much effort to exploit and refine each promising area, as if one would exploit and concentrate only on a single area. Referring again to the stage lightning example in Section 3.4 with *two* spotlights; a local optimum could be a solutions where the right spot optimally illuminates the left half of the stage and vice versa, where in fact the global optimal solution would have each spot illuminate their respective half optimally. The challenge here would be to perturb the solution enough to change both spot orientations and produce a solution with a reasonable fitness. As perturbation generally leads to worse solutions and the heuristic must be able to give such bad solutions a chance to improve. A common analogy is that of a mountain climber who has to get from one peak (i.e. local optimum) to the next. To reach the better peak he first needs to descent to much lower heights before he can again start to ascend towards and explore the new peak. Bear in mind that the algorithm only knows of the landscape it has visited so far, and cannot know whether it is heading toward a higher peak, a plateau, a valley or a trench.

In the context of EAs, mutation and recombination operators may be attributed exploration characteristics whereas the selection operator is contributing to the exploitation behaviour [19]. As also noted this view may be a bit simplistic as operators themselves can be quite complex and parameter settings may produce opposing effects. Also the characteristics of the problem will influence the effect a given operator has in an algorithm which is again related to the "No free lunch" theorem. Another view discussed in [19] originating from [8] characterises an algorithm as exploiting if it is able to step in the direction of desired improvement. Exploration

Algorithm 1 Basic GA

$Initialise(X_0)$ {randomly generated individuals}
$t \leftarrow 0$
$N \leftarrow |X_t|$
while $t < t_{max}$ **do**
  $C \leftarrow \{\}$ {empty population of children}
  for $i = 1$ **to** $N/2$ **do**
    $p_1 \leftarrow f_s(X_t)$ {select two parents}
    $p_2 \leftarrow f_s(X_t)$
    $(c_1, c_2) \leftarrow f_c(p_1, p_2)$ {recombine to yield two children}
    $c_1 \leftarrow f_m(c_1)$ {mutate children}
    $c_2 \leftarrow f_m(c_2)$
    $c_1^f \leftarrow f(c_1)$ {Calculate and assign fitness of child individuals}
    $c_2^f \leftarrow f(c_2)$
    $C \leftarrow C \cup \{c_1, c_2\}$
  end for
  $P_t \leftarrow C$
  $t \leftarrow t + 1$
end while

and exploitation characteristics are valid for any search or optimisation heuristic. For population based algorithms it is closely related to increasing or reducing diversity in the population.

3.5.3 Multiobjective Optimisation

In many real world problems there are often more than one objective to optimise. If for instance you control an autonomous car with one objective of completing a journey as quickly as possible, a second objective could be to keep the fuel consumption low. A problem can consist of an arbitrary number M of objectives and the fitness function defined in Equation 3.5 becomes M dimensional. It is however questionable how practical it is to model tens or hundreds of objectives, as it requires a sophisticated algorithm and makes the result difficult to comprehend for humans.

$$\begin{aligned} F &: \Omega \rightarrow \mathbb{R}^M \\ F(\mathbf{x}) &= \{f_1(\mathbf{x}, f_2(\mathbf{x}, ...f_M(\mathbf{x})\} \end{aligned} \tag{3.5}$$

Returning to the example, a third objective could be that of achieving the highest possible maximum speed. Obviously the first two objectives are in conflict contrary to the first and the third, as a high maximum speed inevitably will lead to a faster time. This problem is then arguably only a bi-objective problem and in such cases redundant objectives should be eliminated at design time.

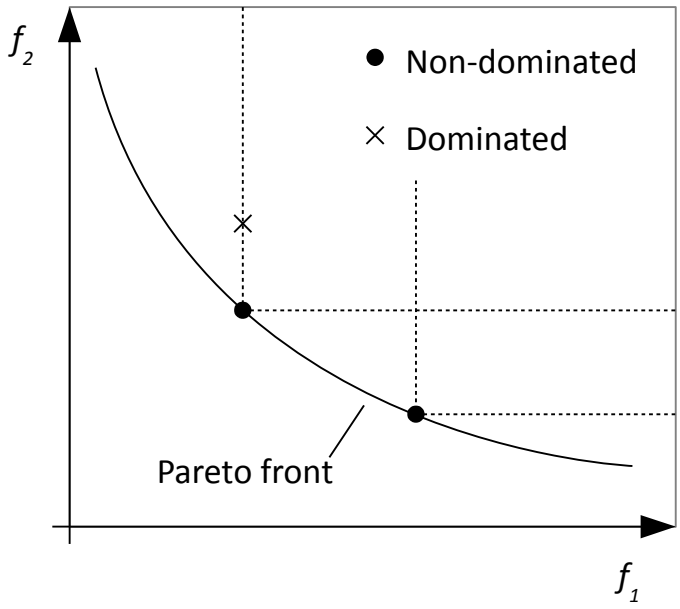


Figure 3.4: Pareto optimality in objective space of objective functions f_1 and f_2 in minimisation.

As stated, the objectives in a multi-objective problem must be in conflict, hence comparing two individuals in terms of their fitness becomes difficult when they are superior in one objective and inferior in others. If the latter is the case it is not immediately possible to define which individual is the best, and both individuals must be regarded as equally good. This leads to the formulation of Pareto domination: An individual $\mathbf{x_1}$ is said to Pareto dominate another individual $\mathbf{x_2}$ if it is equal or better in all objectives, and strictly better in at least one objective. Using this approach the set of Pareto dominating individuals known as the Pareto front can be determined which is illustrated in Figure 3.4 where two dominating solutions of the Pareto fronts are shown with a non-dominating solution.

A widely used multi objective algorithm is the Non-dominated Sorting Genetic Algorithm (NSGA-II) [16] shown in Algorithm 2. The mating and offspring creation process is the same as in the standard GA shown in Algorithm 1, but instead of accepting all children as the new population, offspring and parents are merged in line 15. By sorting according to dominance, the union population is truncated to the size N.

Multi-objective algorithms will usually try to determine the entire Pareto front in the full range of all objectives and let the user select the solution seeming to best correspond to the requirements. This is due to the fact that the shape of the Pareto front may influence the users decision. A solution may be particular interesting if a small gain in one objective leads to a large sacrifice in other objectives.

Algorithm 2 NSGA-II

$Initialise(X_0)$ {randomly generated individuals}
$t \leftarrow 0$
while $t < t_{max}$ **do**
  $C \leftarrow \{\}$ {empty population of children}
  for $i = 1$ **to** $N/2$ **do**
    $p_1 \leftarrow f_s(X_t)$ {select two parents}
    $p_2 \leftarrow f_s(X_t)$
    $(c_1, c_2) \leftarrow f_c(p_1, p_2)$ {recombine to yield two children}
    $c_1 \leftarrow f_m(c_1)$ {mutate children}
    $c_2 \leftarrow f_m(c_2)$
    $c_1^f \leftarrow f(c_1)$ {Calculate and assign fitness of child individuals}
    $c_2^f \leftarrow f(c_2)$
    $C \leftarrow C \cup \{c_1, c_2\}$
  end for
  $U_t \leftarrow X_t \cup C$ {create parent + offsrping union of size 2N}
  $F \leftarrow fastNonDominatedSort(U_t)$
  $X_t \leftarrow truncate(F)$ {based on domination and crowding}
end while

3.6 Summary

A brief introduction to the historic field of search heuristics was presented with a closer look at evolutionary algorithms. The origins in bio-inspired computation were underlined with a simple example of optimisation problem and a generational Genetic Algorithm (GA). Challenges and goals in evolutionary algorithms were discussed and the chapter was rounded of with a description of the extension of the single-objective problem to a multi-objective one. The terms Pareto front and dominance were defined and as an example of a multi-objective algorithm the Non-dominated Sorting Genetic Algorithm (NSGA-II) was outlined.

4

Literature Overview in Protein Design and Inverse Folding

4.1 Introduction

The structure of a protein is typically represented by different levels of structures (see Fig. 2.2). The primary structure is the protein sequence of N amino acids $\{aa_i\}$ where $1 \leq i \leq N$ is the residue position. The secondary structure defines the organisation of *helices*, *sheets* and *loops* of the tertiary structure and can be expressed by a type $\{T_i\} \in \{H, E, L\}$ for each position i in the protein. The tertiary structure completely describes the arrangement of all atoms of a single sequence in the three-dimensional space. A simplified example is presented in Fig. 2.2 with only N and C atoms comprising the protein backbone structure and R_i residue side-chains. The ensemble of three-dimensional positions of C_α atoms is commonly referred to as the alpha-trace which provides a rough residue type- and rotation-independent view of the proteins organisation.

4.2 Comparing Structures

When working with proteins and their structures it is often required to compare proteins both visually and quantitatively on different levels of their structure. This begins at the level of the primary structure, comparing one sequence to another or to a database of sequences. A crucial part here is to identify sub-sequences common to the sequences being compared even if these do not have the same lengths. Naturally gaps then need to be taken into account and the problem is often referred to as sequence alignment which is seeking to identify distant ancestry among protein sequences. The types of amino-acids are also taken into account during alignment as some mutations are more likely to occur through natural mutation than others. This is however outside the scope of this work where sequences are always of the same length and inherently aligned to the target structure except when comparing final

structures in the experimental validation sections.

In terms of tertiary structure comparison, various scoring functions exist. A widely used and simple measure is the Root Mean Square Deviation (RMSD). It is based on the pairwise distance between residue positions in the two tertiary structures to be compared. With $S^a = \{s_1^a, s_2^a, ...s_N^a\}$ and $S^b = \{s_1^b, s_2^b, ...s_N^b\}$ denoting the 3D positions of every residue in the two structures to compare, the RMSD is defined in Equation 4.1, assuming the structures are optimally aligned.

$$RMSD(S^a, S^b) = \sqrt{\frac{1}{N'} \sum_i |s_i^a - s_i^b|^2},\ i \in \{i|\ |s_i^a - s_i^b| < 5A\} \tag{4.1}$$

If a perfect alignment exists, the sum of deviations would be zero. The unit of measurement on the molecular scale is $1A = 10^{-10}m$ and in practice, values below $1A$ comparing two different proteins are rare. Existing methods seek to align the largest possible fraction of the structures or longest continuous segment, such as the Local and Global Alignment (LGA) tool detailed in [76]. The RMSD score reported is taking only the residues within $5A$ cut-off into account, hence focusing on parts that are similar and ignoring dissimilar regions which would otherwise mask the similar regions. The Global Distance Test (GDT) reported by the tool is a measure indicating the average of percentage of residue positions that can be fitted below each of the thresholds $\{0.5A, 1.0A, 1.5A, ...10.0A\}$. The GDT Total Score (TS) averages percentages in $\{1.0A, 2.0A, 4.0A, 8.0A\}$.

The Template Modeling Score (TM-Score) proposed by Zhang and Skolnick in [77] is another sequence independent score that seeks to address shortcomings of other similarity scores by incorporating the entire structure in the calculation. A later study by Xu and Zhang described in [74] demonstrates the significance of the TM-Score and relates score values to the likelihood that the tested structures are actually in the same fold.By computing P-value for different TM-Scores, it is found that a TM-Score above 0.5 indicates that the tested structures are generally in the same fold.

4.3 Protein Design

Most applied work of the IFP is concerned with protein design. Since the first design of a peptide by Gutte *et al.* [27] using secondary structure rules, numerous works have described different approaches to the IFP problem. The earliest reference to the inverted approach is found in an article by Pabo [52] referring to Drexler [18]. He stated that protein design engineers could in theory choose from a vast subset of possible sequences containing strategically placed groups that would have a predictable fold. Another early attempt at tackling the IFP is done by Ponder and Richards [55] who used a systematic exhaustive approach of enumerating a selected subset of residue positions. Central to the approach is the focus on packing criteria of core residues, taking a latest available side-chain rotamer library into account. Core

residues are internal or buried residues not in contact with solvent. They contribute to the general structure of the protein and rather seldom to its primary function. A rotamer-library is a library of known side-chain arrangements in 3D for each residue which is important to consider when evaluating the space filling of the core structure.

A few years later, Bowie *et al.* [9] introduced a 3D to 1D score for each secondary structures type and six environmental classes determined by 1) area buried in the protein structure and 2) fraction of polar side-chain area. By analyzing 16 known structures the overall relative probability of observing a residue in a defined environment class is computed. From this and the target tertiary structure, a 3D profile can be generated taking the environment at each residue position into account. The 3D to 1D score is calculated by matching a sequence to the 3D profile of a structure. The result is expressed relatively using the Z-score, indicating the number of standard deviations above the mean of other sequences of same length. Using this method they were able to clearly separate homologs (evolutionary related proteins) in terms of Z-score from a large set of sequences.

Kuhlman and Baker [40] used a Monte Carlo approach of residue and rotamer substitution at 11 non-adjacent core positions, evaluating a free energy function. The lowest energy sequence of five algorithm runs was chosen and as a final result half of the generated residues were identical to the reference protein, referred to as 'wild-type'.

The first to use a Genetic Algorithm (GA) was Jones in [33]. To assess 3D-1D compatibility and define an objective function, a set of statistically determined potentials known from fold recognition were used: pairwise potential and solvation potential. To prevent the generation of unlikely sequences, a residue composition term with an arbitrary weight was added, that corresponds to the target folding class ($\alpha\alpha$, $\alpha\beta$, $\beta\beta$). Jones concluded that there was no way to be sure the resulting sequences have not been over-designed as the optimal sequence scores significantly better than the reference. He speculates that the energy optimal shape might be very steep and too hard for the real-world protein to fold into. Therefore the algorithm should possibly be stopped earlier.
Mayo *et al.* [62] successfully used backbone flexibility in the design process by generating a set of perturbed backbones. To cut the search space, only 10 residue positions were enumerated and Dead-End-Elimination applied. Similarly, Harbury *et al.* [28] incorporated such backbone freedom in their design approach. Both latter approaches were evaluated by synthesizing the proteins in the lab. Isogai *et. al* [31] used a recursive approach searching the 3D profile of the target structure, by keeping two residues fixed and applying a penalty to residues that protrude into the space with a repulsive function. Bumps among side-chains were removed manually by replacing residues with smaller ones. The design was successfully synthesized, but the binding site did not stably bind oxygen.

Wernisch *et al.* [69] sought to combine the latest approaches into an automatic software solution named DESIGNER. The CHARMM package [10] was proposed

for force-field calculation among side-chains and backbone, taking all hydrogen atoms bonded and non-bonded into account, as well as adding van-der-Waals forces and electrostatic interaction. Both an exact branch-and-bound approach and a simple heuristic selecting the optimal rotamer for one random position at a time until a local optimum has been reached, were tested. Different experiments were conducted comparing the effects of different settings on the results. One test compares the effect of neglecting the reference energy and solvation energy terms respectively when redesigning 11 buried positions in the core. The choice of energy terms largely impacted the amount of polar amino acids, and neglecting the solvation term produced better packing with less cavities. Another test aimed at optimising the protein surface with its larger proportion of polar amino acids. Again 11 positions were variable and varying settings were tested. First backbone and rotamers were kept fixed, then alternative rotamers were allowed. Wernisch *et al.* considered that the energy calculations were approximations. Therefore the software allowed for outputting multiple solutions within a user defined energy window. Where packing constraints apply, DESIGNER generated sequences close to the reference.

Voigt *et al.* [68] combined the field of directed evolution with that of computational design and sought to benefit from both. Directed evolution is concerned with improving specific protein properties or functions mainly by applying a series of mutations to the target as mutagenesis in nature. In their computational method, energy was used to predict structural stability and residues with low entropy are detected as more tolerant to mutations. They also argued that coupled residues should be substituted together as several replacements need to take place to demonstrate improvement. High variability was observed on the exposed residues, and in general the variability should guide mutagenesis to allow the generation of a family of divergent sequences with structural integrity intact.

Klepeis *et al.* [38] presented a two stage approach where an integer program was first used to generate a list of low energy sequences which were then evaluated in terms of their fold. Using a force-field based on pairwise C_α distance dependent interaction potential gave a more relaxed backbone flexibility constraint with less empirically tuned parameters. Validation was done by improving the activity of Compstatin, a 13 residue long peptide fundamental in inhibiting complement activation. Certain residue positions and types were restricted based on knowledge about the functional nature and with the goal of increasing activity. Experimental results on 14 designed sequences showed significant activity improvements in most cases, one analogue was 6-7 times more active than the wild-type underlining. This two stage approach with small variations was used to design a template for human β-defensiv-2 in [22] and with more advanced second stage in [6] and [5].
Smadbeck *et al.* [61] have recently streamlined the two stage process and present a server implementation with a usage example. The web-interface allows for specifying all inputs: template (rigid/flexible), energy function (C_α, centroid or any) and biological constraints (on charge and content). Stage two work-flow consists of two independent fold specificity and approximate binding affinity modules. These in-

clude programs such as CYANA [26], TINKER [54] and AMBER [13] for the first, Rosetta [58] (-Abinitio, -dock and -design) and OREO for the latter.

Finally, Mitra *et. al* [46] used templates of structure families in combination with a force-field to guide the search rather than physics-only based force-fields. Due to the shortcomings of the latter, evolutionary based designs have been demonstrated to be more stable. Experiments were conducted with one of the leading protein structure prediction frameworks, I-TASSER[75]. Previous works have shown that I-TASSER is able to distinguish successful designs from unsuccessful ones and is therefore used as validation of the optimisation algorithm results, likewise in this work. The works discussed in this chapter are summarised in Table 4.1 with a brief description.

4.4 Conclusions

In summary, the efforts in solving the IFP over the last three decades have been focusing on finding few highly refined solutions to the problem. The final output of these methods consists of a single or few sequences close to the input sequence, mostly applied for protein design purposes or improved docking, where a larger set of different sequences can be desirable by practitioners. Little recent efforts have strived to obtain such larger collection of possible solutions, or to explore characteristics and features of larger solution sets. In addition, existing works often target a subset of the sequence, to reduce the space of possible solutions. This choice is related to the fact that these methods employ enumerative or exhaustive sampling of the solutions space which is not feasible with long protein sequences. The use of costly all-atom force field approximations further impedes full length sampling of many sequences. Such a sampling on the full sequence length seems to be a very challenging problem suited for a meta-heuristics search approach such as Evolutionary or Genetic algorithms, and to the knowledge of the author, no recent works have attempted to tackle this problem.

Author	Algorithm/ heuristic type	Goal	Model components/ features	Validation method
Ponder and Richards[55]	EE	M	RL, PC	Preliminary tests
Bowie, Lüthe and Eisenberg [9]	EE	M	RAB, FPS	Z-score on homologs
Jones [33]	GA	DPD	EC, ES, CC	Threading onto decoy folds, energy observed
Mayo and Su[62]	DEE	DPD	BF, EVNW, ES	Experimental on 7 samples
Harbury et. Al [28]	EE	DPD	BF, RL, PC, SE	Experimental on 4 samples
Isogai et. al [31]	EE	DPD	PC	Experimental on 1 sample
Wernisch et al.[69]	BB/H	DPD	RL, EC, ES	Cavitation analysis and comparison to wild-type
Voigt, Mayo, Arnold and Wang [68]	DE, EE	DPD	RL, AFF	Comparison to previous designs by directed evolution
Kuhlman and Baker [40]	MC	C	AFF	Comparison to wild-type
Klepeis et. Al [38]	BB, EM	DPD, BSD	2 stages, BF, AFF	Experimental on 1 sample, with improved activity
Fung et. Al [22]	BB, EM	DPD	2 stages, BF, AFF	Experimental on 1 sample
Smadbeck et. Al [61]	BB, EM	DPD, BSD	2 stages, BF, AFF	Theoretical
Mitra et. al [46]	MC	DPD	AFF+ST/AFF	I-TASSER validation

EE: Exhaustive enumeration
GA: Genetic Algorithm
DEE: Dead-end-elimination
BB: Branch-and-bound
H: Heuristic
EM: Energy minimization
MC: Monte Carlo
DE: Directed evolution

M: Match known sequences to known structures
DPD: Denovo protein design
BSD: Binding site design
C: Conclusion about sequence to structure match

RL: Rotamer library
PC: Packing constraints
BF: Backbone flexibility
EC: Conformation energy
ES: Solvation energy
EVNW: Van der Waals
RAB: Residue area buried
FPS: Fraction polar side-chain-area
AFF: Advanced force field
CC: Composition constraint
SE: Stability estimate

Table 4.1: Table of algorithms and approaches in protein design

5

Literature Overview in Niching and Diversity Preserving Techniques

5.1 Introduction

In metaheuristics, the subject of exploration vs. exploitation characteristics has been thoroughly studied and discussed (see Section 3.5.2). For population based optimisation algorithms it is well-known that a higher level of population diversity results in more exploration at the expense of exploitation. With a set limitation on the evaluation budget this represents a choice of whether to refine the solution(s) already found, or continuously searching new areas of the solution space for possibly even better solutions. In general, if diversity approaches zero it indicates that the algorithm has converged towards a single solution, which might be an undesired behavior if it occurs too early during the algorithm execution. In situations of *premature convergence* the algorithm is effectively stuck in a local optimum from which it has difficulties of escaping. The nature of the fitness landscape has a large influence on this aspect, if for instance the landscape is very rugged with many deceptive local optima, high diversity and exploration is desirable. On the other hand, if the fitness landscape is smooth consisting of a single global optimum, exploitation would be desirable. This is what could be expected according to the No Free Lunch Theorem mentioned earlier. Many approaches have been designed to address and influence the diversity of the population. The most prominent works will be presented and discussed in the following

5.2 Problem Characteristics

Whether working with continuous or discrete optimisation problems, preserving some degree of population diversity is generally considered to be an advantage. Especially

in the case of *multi-modal*, *deceptive* and/or *dynamic* problems. The following sections discuss relevant works in landscape analysis and problem feature characterisation. The aim is also to underline the importance and influence of diversity and niching features in GAs when tackling such problems.

5.2.1 Dynamic Optimisation Problems

A *dynamic* optimisation problem (DOP) is characterised by an objective function that changes over time. In many real-world scenarios the optimisation problem at hand can be influenced by external events of potentially unpredictable character. For instance the optimal routes of a delivery service may be influenced by traffic jams, a sudden failure of a van, incoming orders of high importance etc. Like natural species in a changing environment, the population must be ready to adapt to a new fitness landscape when a change occurs. If the individuals are highly diversified, chances that a subset will be fairly well adapted to the modified environment are higher. On the other hand if the individuals are similar and highly specialised, the risk of them all failing in the new environment is higher. The type of changes in the environment can be either linear (gradual) over time or abrupt. Often the change is repetitive with a certain frequency and a certain severity. In a linear changing environment, the optimal solution continuously moves around the search space. Likewise for the abruptly changing landscape an optimal solution may be replaced by another, but catastrophic changes where the entire environment is suddenly replaced by a new one are also tested in literature. Further the changes may impact the overall quality of solutions or constraints defined on the problem. The topic has received a lot of attention by the evolutionary computation community early on. In [12], Cobb and Grefenstette cover many relevant approaches and compare performance of three: *Random immigrants*, *Triggered Hypermutation* and high mutation in a standard GA. The latter performed well for linear changing environments, but as for the other two approaches the increased randomness comes at the cost of reduced overall quality. *Triggered Hypermutation* has the advantage of sensing the change and reacting to it, but though it is better performing overall it also has to have its parameters set according to the problem. Apart from adaption, diversity or elevated mutation is identified as a key feature for efficiently tracking the changing optimas. In a more recent work [20] Farina, Deb and Amato, provide an initial study of the multiobjective case where a changing Pareto-front needs to be tracked in time, rather than individual optimal solutions. Existing test problems are modified to become time dependent and a real world example of a dynamic proportional-integral-derivative (PID) controller for a rubbish burner is presented. They conclude that a key algorithm feature is to produce a sudden increase in diversity to get out of a converged set of solutions and that their study only just scratches the surface of the topic. The interested reader can find a very complete review of benchmarks, performance measures and methods for dynamic optimisation problems in [50].

5.2.2 Deceptive Optimisation Problems

A *deceptive* problem is characterised by having seemingly good locally optimal solutions which attract the algorithm away from the optimal solution(s). Based on the initial work of Bethke in [7] referring to the type of problems as *misleading*, Goldberg presents a formal definition of the *deceptiveness* of problems in [23]. Attractors are the convergence points that genetic operators of a GA will ultimately attempt to reach. If these points are global optima Goldberg defines the problem as *simple* otherwise it is *deceptive*. At the time of the work of Mitchell, Forrest and Holland [45], deceptiveness was regarded as one of the leading features that made optimisation problems difficult for GAs. The authors define additional features such as isolated high fitness regions, multiple conflicting solutions and hierarchical structured regions. The first feature of isolated fit regions is closely related to deceptiveness, but the latter two hint at the multi-modal landscape feature discussed in the next section. In fact a problem can easily be both *deceptive* and *multi-modal* as noted in [24], and the two features are closely related.

5.2.3 Multi-modal Optimisation Problems

A problem can be characterised as *multi-modal* if it has many local optima with comparably equal fitness. There may also be more than one global optimum, but it is not a requirement. Hence a *multi-modal* problem can be characterised by the amount of peaks, i.e. local optima, their variation in fitness and their distribution in the search space. Each of the optima attract the algorithm and without countermeasures it will concentrate solutions around few optima. Apart from having many widespread local optima that require elevated diversity to be explored, *multi-modal* problems present an additional challenge: That of preserving the best individuals found so far. With a diversified population the probability of retaining more local optima is higher. This will at the same time increase the likelihood of the global optimum or optima of being represented as well. Depending on the purpose it may even be desirable to find all or as many as possible of the local optimas, which in the nature of the problem will have very similar fitness. When testing algorithms the performance is hence often reported as peak (i.e. local optima) ratio, diversity and average fitness. Section 5.4 discusses numerous niching methods in-depth of which the majority are designed to target *multi-modal* problems.

5.3 Diversity Measures

There are several ways of defining and measuring diversity in a population of individuals. Some works apply diversity measures in the objective space to fitness values, others, as in this work to the individual's loci in the solution space. To measure diversity, first a distance metric is required that can indicate to what extend two solutions are different. To produce a good indicator it is important that the metric satisfies that

the proportion of change in distance is proportional to the change in solution *phenotype*. The nature of such a distance metric is crucially impacted by the type of solution encoding, i.e. using real or discrete values. For real value encoded solutions an extensive study can be found in [14]. In this work the focus is on discrete solution encoding, and a few important discrete diversity measures are discussed in the following.

5.3.1 Pairwise Hamming Distances

For discrete and binary coded solutions, the Hamming distance between two solutions is defined as the number of loci that contain different values. In other words, the number of loci in the solution that need to be changed in one solution in order to transform it into the other. To measure the diversity in a population, the Hamming distance between any pair of individuals in the population is averaged:

$$H = \sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} d_H(x_i, x_j). \tag{5.1}$$

where $d_H(x_i, x_j)$ is the Hamming distance between the pair of individuals x_i and x_j. Hamming distance based diversity measures are the most widely used in discrete coded solution spaces and the easiest to grasp and interpret.

5.3.2 Information Entropy

Entropy is known from thermodynamics and was later used in 1948 by Claude Shannon in communication theory. It can be regarded as a measure of the amount of information and it also has its relevance in compression algorithms among others. In [65] the entropy H_i is defined per locus i:

$$H_i = -\sum_{j=1}^{N} \frac{n_{ij}}{N} ln \frac{n_{ij}}{N} \tag{5.2}$$

where n_{ij} is the number of times the j_{th} value occurs at locus i. The maximum value of H_i is the population size N if each of the j values occur uniformly in the population at locus i and 0 if one value dominates.

5.3.3 Moment of Inertia Diversity Measure

Another measure is presented in [47] and borrows ideas from engineering and moment of inertia.

$$I = \sum_{i=1}^{N} \sum_{j=1}^{P} (x_{ij} - c_i)^2. \tag{5.3}$$

where x_{ij} is the i_{th} position in the j_{th} individual and c_i corresponds to the centroid or average of all values at position i. The measure works for both real and discrete encoded solutions. In addition it produces equivalent results as the pairwise Hamming measure on binary coded solutions, even though it is computationally more efficient.

5.4 Niching Approaches

De Jong was the first to introduce the concept of niching in [15] supposed to emulate the natural speciation known from nature. Different species have to compete over a limited number of resources and will eventually specialise and coexist in their own niches. The purpose and advantage being to enhance the exploratory characteristics of the algorithm, rather than having it converge to a population with identical highly fit individuals.

5.4.1 Crowding Methods

The standard crowding method proposed by De Jong [15] uses a mechanism limiting the number of individuals occupying the same area of the search space based on their genotype. Evolving only a part of the individuals corresponding to the *crowding factor* CF, the generated offspring replaces the most similar individuals in the population. This mechanism prevents similar individuals from coexisting. Mahfoud [43] analyses a number of relevant crowding methods by measuring peaks maintained and replacement errors produced when inserting offspring individuals into the population. The number of replacement errors of other methods are found to be significant, and Mahfoud combats this by introducing *deterministic crowding*. Here, the generated offspring competes only with their parents for replacement and Mahfoud reports good results. Another example where individuals are selected for survival, or elimination respectively is Shimodaira's approach [60]. After mating has taken place, identical individuals and the less fit half of the merged parent-offspring population are eliminated. Then survivors are selected for the next generation based on the Hamming distance to the best individual. This essentially eliminates individuals which are too similar to the best individual. If too few individuals remain, random individuals are created and inserted.

5.4.2 Fitness Sharing and Clearing

Fitness sharing and *clearing* are two other approaches closely related to *crowding*. The idea is also to limit the density or the number of individuals that may populate a given area of the search space by emulating resource limitations of the species in a niche. *Fitness sharing*, first introduced by Goldberg and Richardson in [25], achieves this by letting individuals in a niche share fitness. An individual will have to share fitness with all other individuals within the *sharing radius* according to a *sharing function* that penalises closer individuals more. The *sharing function* can

be anything, but initially a power function was used. As the lower fitness of sharing individuals reduces their chances of being selected for mating, even in high fitness areas, the approach prevents many individuals from crowding a single high fitness area, niche or peak. With a limited population size, this on the other hand leaves space for individuals with low fitness in remote areas to survive and discover other niches. In *clearing* introduced by Petrowski in [53], dominant individuals are identified as representatives of each niche and all members of a niche are found by checking whether they fall within the *clearing radius*. Eventually all dominated individuals of a niche will have their fitness reset before the next round of mating, which basically eliminates them. Laredo *et al*. [41] proposed altering the fitness of individuals selected in a cooperative tournament selection scheme based on altruism. The fitness of selected individuals will be reduced, lowering the chances of being selected multiple times, hence preserving diversity.

5.4.3 Sequential Niching

Sequential niching is an approach where numerous GA runs are launched one after the other. Where a basic GA does not implement learning, knowledge of the found optimal peaks is carried over from previous GA runs to the next in [4]. This information is then incorporated in the fitness landscape similarly to the methods of fitness sharing. The general sequential approach however has some drawbacks and Mahfoud [44] concluded that parallel or sharing based methods perform better. In [67] Vitela and Castanos proposed a hybrid sequential memetic algorithm combining local search with a GA that alternates between exploration (diversification) and exploitation (intensification) phases. By storing the local optima found, the algorithm employs fitness sharing and clearing mechanisms to discover more optimal solutions with each new sequential iteration.

5.4.4 Hierarchical Niching

Apart from introducing a hierarchical niching optimisation model in [30], Hu and Goodman discussed the concept of *spatial* and *temporal* niching. The latter consists in the sequential approaches introduced in Section 5.4.3. *Spatial* niching comprises the remaining methods where niches are preserved in parallel in the population. The hierarchical approach allows the coexistence and coevolution of individuals with different fitness levels and lets offspring migrate up in the hierarchy if their fitness level qualifies them. The aim is to address the difficulty experienced by a specialised or converged population to retain individuals of new less fit regions. The authors described this niching technique as both *spatial* and continuously *temporal* at the same time.

5.4.5 Structured Algorithms

Since the earliest works on evolutionary algorithms the idea of structuring the population has been studied and applied as discussed in [2]. Structuring can happen at many levels, from coarse-grained parallel implementations known as the *island model* to fine-grained cellular algorithms. The *island model* can hardly be attributed to a single author as the idea of running GAs in parallel was conceived simultaneously with the invention of GAs. Parallel execution is used just as much to achieve speed-ups on multi-processor hardware, if not more than, for diversity preservation. A specific study of the island-model mentioning diversity preservation characteristics is presented by Whitley, Rana and Heckendorn in [71]. They show that the isolation provided by islands islands are in fact a source of diversity. Araujo and Merelo [3] introduce a migrant selection mechanism to improve diversity in an island model. It is based on genotypic differences of the immigrant individual and the destination subpopulation in which it is to be included. Cellular algorithms [1] seek to maintain niches and diversity by means of a structured neighbourhood within the population. Cells are arranged on a grid each containing an individual which is only allowed to interact with its neighbours on the grid. The neighbourhood definition can be modified to include more or less cells impacting the diversity preserving effect.

5.4.6 Multiobjectivization

A different approach to preserving diversity and niches is that of extending the original optimisation problem with an additional objective. This method is referred to as *multiobjectivization*. The initial purpose of this technique proposed by Knowles in [39] was to prevent the algorithm from getting stuck in local optima. This was done by transforming the fitness landscape and reducing the amount of local optima, by extending or replacing the original objectives. The idea being that the algorithm would have to be stuck in all objectives at once to be truly stuck. Logically the optimal solutions of the modified problems fitness landscape need to be the same as in the original problem. Further, the developed *helper-objective* must be in a conflict with the primary objective of some sort, otherwise the resulting problem cannot be regarded and tackled as a multi-objective one. Using NSGA-II as a base-algorithm, Jensen [32] created helper objectives to solve job shop scheduling problems (JSSP) minimising the total flow time where the helper objective was defined as minimising flow time of individual jobs. These objectives may not seem conflicting, but minimising flowtime of one job will most likely come at the expense of delaying at least one other job. Creating as many objectives as there were jobs proved to be too many, as it removed the selection pressure or focus away from the original objective. Hence fewer objectives were switched dynamically between the jobs. There are many ways of designing the additional objectives, but Toffolo and Benini [64] were the first to specifically apply an objective for genetic diversity in their Genetic Diversity Evolutionary Algorithm (GDEA). The genetic diversity objective is defined as the minimal euclidean distance to any other individual in the population. With a slightly modified

dominance definition and removal of clones (identical individuals) the GDEA outperformed NSGA and SPEA on the six test functions of the ZDT test suite [78]. Deb and Saha [17] specifically sought to achieve *niching* in *multi-modal* problems. Additional objectives designed included, a gradient objective and a neighbouring solution count. The rationale behind minimising the objective function absolute gradient is that this should preserve the local optima found by the algorithm as they will have zero slope in theory. Minimising the number of solutions in a defined neighbourhood is similar to maximising the diversity. Employing a modified dominance as in [64] and a clearing method, successful results solving multi modal test problems were achieved. Wessing *et al*. [70] continued in the same mindset employing nearest better neighbour information to design an additional objective. The non-dominated fronts emerging from the additional objectives are sorted according to the original objective or the designed diversity-objective to focus more or less on the diversity objective. By iteratively removing the worst individual, the combined parent and offspring population is reduced until the best individuals have been selected for the next generation. These latter works have specifically shifted attention towards *multi-modal* problems, and seek to find and preserve all optimal/local solutions.

5.5 Conclusions

Diversity in evolutionary algorithms has since the very beginning been a subject receiving a lot of attention. Mainly due to the fact that an evolutionary algorithm population without diversity only evolves very slowly and hampers its ability to explore new areas of the search space and escape a local optima. Many works have addressed aspects regarding diversity, such as measuring, analysing and controlling it. This chapter gives a brief overview and background of topics related to niching and presents related works that have employed many different methods specifically to maintain diversity. These related works are summarised in Table 5.1. It is clear that there are a lot of ways to achieve diversity, both with complete new mechanisms and alterations to existing algorithms.

The Diversity as Objective (DAO) approach seems to have the highest potential to suit the requirements of the problem description. On one hand it is diversity preserving and on the other, diversity is directly formulated as an objective. This should allow the algorithm to output a collection of good solutions rather than converge to a single solution. Further, no work applying DAO approaches seem to provide a means of gradually dosing the amount of diversity induced, nor do they apply DAO to problems with discrete encoding. This opens the opportunity to contribute with a novel use-case and new algorithm features.

Author	**Method**	**Details**
De Jong [15]	De Jong crowding	replace most similar
Mahfoud [43]	Deterministic crowding	replace worst parent
Shimodaira [60]	Diversity control	eliminate similar
Goldberg and Richardson [25]	Fitness sharing	Sharing function for fitness degradation
Petrowski [53]	Clearing	Clear fitness within clearing radius
Laredo *etal*. [41]	Cooperative tournament selection	Altruism
Beasley [4]	Sequential niching	Fitness sharing
Vitela and Castanos [67]	Sequential niching	Hybrid of GA and search. Fitness sharing and clearing
Hu and Goodman [30]	Hierarchical niching	Spatial and continuous temporal niching
Araujo and Merelo [3]	Island model	migrant selection
Alba [1]	Cellular algorithm	fine-grained structure
Toffolo and Benini [64]	Multiobjectivization	diversity as objective
Deb and Saha [17]	Multiobjectivization	diversity as objective
Wessing *etal*. [70]	Multiobjectivization	diversity as objective

Table 5.1: Table of algorithms and approaches

Part II

IFP Problem Modelling and Algorithm design

6

Model Presentation

6.1 Introduction

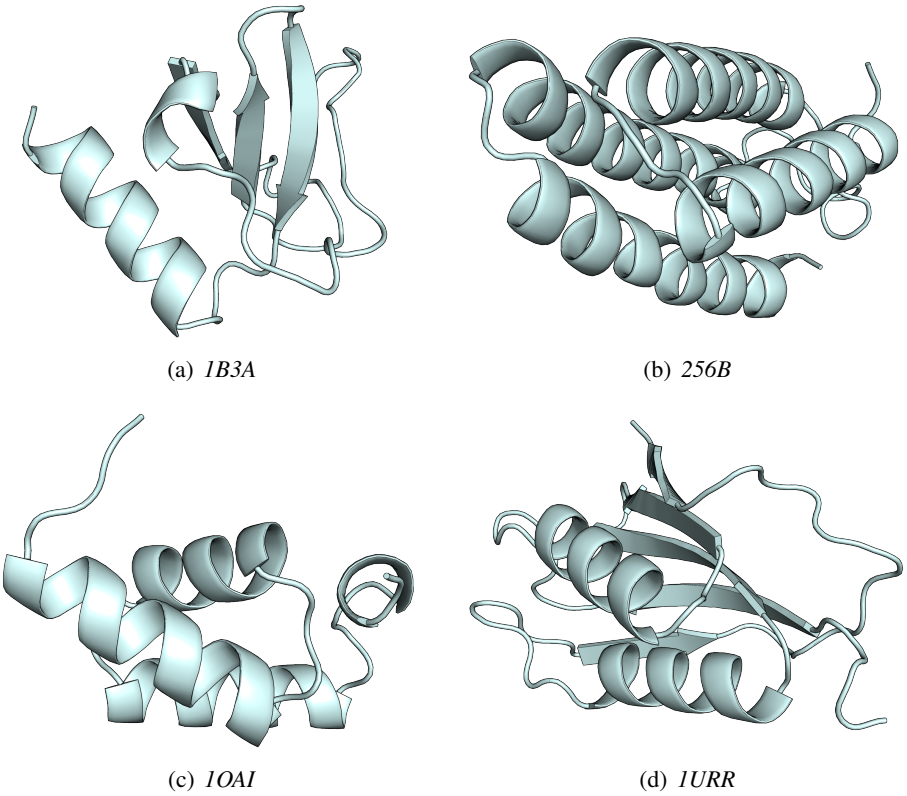
The focus in this work is on finding multiple and diversified solutions to the Inverse Folding Problem (IFP) according to the problem description in Chapter 2. In the following, a simplified model is developed to matching solely the reference secondary structure - a requirement for the tertiary structure, see Fig. 6.2 for a schematic representation. This is motivated by the fact that computing the tertiary structure of a given input sequence is computationally very expensive which would prevent the usage of a metaheuristic on the entire sequence. This chapter introduces the four proteins that were chosen as problem instances as well as definitions and evaluation methods of the model.

6.2 Protein Samples

The four protein samples chosen, namely *1B3A*, *256B*, *1OAI* and *1URR* are illustrated in Fig. 6.1 (a-d).
1B3A named *"Rantes"* is a 67 residues long protein with 2 helices and 3 sheets. It can inhibit, i.e. prevent, the entry of human immunodeficiency virus-1 (HIV-1) into blood cells.
256B named *"Cytochrome b562"* consists of 4 helices and 106 residues. Cytochrome b562 is a variant of Cytochrome b, a protein that plays an important role in the transport of electrons through membranes of living cells.
1OAI connects other proteins as part of a *"Complex"* involved in the transport of molecules to and from the cell nucleus. It is the shortest sample with a length of 59 residues, consisting only of 4 helices and no sheets and hence expected to be the easiest protein to predict.
1URR named *"Acylphosphatase"* is an enzyme, that is, a protein that acts as a catalyst (i.e., helping or accelerating a chemical reaction). It can be found in muscle cells where it is involved in catalysing the production of phosphates as a part of the regulatory processes in the cells. The protein is 97 residues long and consists mainly

of sheet structure with 5 sheets and only two helices. Due to the predominant sheet structure and the length, this protein is expected to be the most difficult to predict.

All in all the samples represent a selection of short to medium long proteins with various structural configurations and functions. These provide enough variety to study the influence of length and structure type aspects on the computational results.



(a) *1B3A* (b) *256B*

(c) *1OAI* (d) *1URR*

Figure 6.1: Three-dimensional structure of the samples

6.3 Definitions

In the following a brief introduction to terms and definitions related to the problem is provided. These definitions will provide a formal ground for the model presentation.

6.3.1 Sequence Representation

In nature, the protein sequence consists of an RNA code and can easily be represented as a sequence of elements. A single solution is represented as a sequence $A = \{aa_i\}$ composed of N residue positions, where $1 \leq i \leq N$ and $aa_i \in \{1,2,...20\}$ corresponds to the set of 20 amino-acids regarded in this work, see Table 2.1. As the solution

space consists of a total of 20^N different combinations, it is clearly not feasible to probe each of them for larger N. Typical design targets, mostly consist of 50 - 200 amino acids yielding an effective search space of $1.13e+65$ for the shortest ones.

6.3.2 Sequence Identity

Sequence identity is a common measure designed to assess the similarity of proteins occurring in nature in terms of their primary structure only. When computing sequence identity, gaps are taken into account during the alignment of the sequences to be able to detect evolutionary relations among the compared proteins even if their sequences are of different lengths. The identity is reported in % often subtracting a penalty for the gaps taken into account to perform the sequence alignment. In this work, all sequences being compared have the length of the target sequence, and are generated by a random process. The chances of the same sub-sequence to occur in two different sequences with an offset diminishes quickly as the sub-sequence length increases, which justifies ignoring gaps in the model. Therefore the Hamming-distance, defined as the number of permutations necessary to convert one into the other is used in the definition of identity. Not taking gaps or varying sequence lengths into account, for two sequences $A = \{aa_i\}$ and $A' = \{aa'_i\}$ where $1 \leq i \leq N$, the Hamming distance between them is defined as:

$$d_{Hamm}(A,A') = \sum_{i=1}^{N} d_i,\ d_i = \begin{cases} 0 & \text{if } aa_i = aa'_i \\ 1 & \text{otherwise} \end{cases} . \tag{6.1}$$

To obtain the sequence identity measure in percent, one would have to write $100 - \frac{100}{N} d_{Hamm}(A,A')$. For the comparison of final results the generally accepted approach with taking gaps into account is used.

6.4 Secondary Structure Definition

Secondary structure refers to the annotation of structure segmentation as seen in Figures 2.2 and 6.2. These segments are the result of the protein naturally folding so that different parts of its 3D structure connect through bonds between amino-acids on separated residue positions in the sequence.

Tertiary structure annotations are done using the 'Define Secondary Structure of Proteins' (DSSP) tool [34]. Though more variants exist we reduce As only the three structure types, *Helices (H)*, *Sheets (E)* and *Loops (L)* are considered throughout this work, some simplification is required.

6.5 Objective Functions

Objective functions are needed to make structure optimisation possible with an evolutionary algorithm. In the following, objective functions are defined to evaluate how

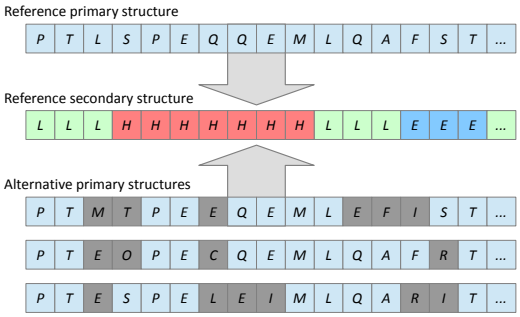


Figure 6.2: Primary and secondary structure in the inverted folding problem

well a given solution fulfils the problem specific requirements: Estimated structure similarity and diversified solutions.

6.5.1 Secondary Structure Estimation

The goal of this objective is to distinguish sequences by assigning a better score to sequences that may match the reference structure better. The model relies on matching solely the reference *secondary* structure as this is a requirement for the tertiary structure though not a guarantee.

Using a secondary structure prediction tool, such as PROFphd [59], the likely secondary structure can be predicted. This makes it possible to probe several sequences per second on any recent computer, contrary to the many hours it would take to compute the full tertiary structure of a given sequence. The resulting prediction consists of 2-tuple with type and reliability estimate at each position in the protein sequence passed as input. The secondary structure type is formally written as $T_{pred}(i)$ per position i and the type prediction reliability $R_{pred}(i) \in \{0...9\}$, likewise for each position i in the sequence. Based on the two vectors T_{pred} and R_{pred} the estimated similarity score $F_{sec}(A)$ is calculated as a sum of matches and mismatches weighted by the reliability:

$$F_{sec}(A) = \frac{\Sigma_{max} - \sum_{i=1}^{N} s_i \cdot (C_{pred}^R + R_{pred}(i))}{\Sigma_{max}} . \tag{6.2}$$

where

$$s_i = \begin{cases} 1 & \text{if } T_{pred}(i) = T_{ref}(i) \\ -1 & \text{if } T_{pred}(i) \neq T_{ref}(i) \end{cases}$$

and

$$\Sigma_{max} = (C_{pred}^R + \max R_{pred}) \cdot N$$

Equation 6.2 is normalised by the maximum possible sum, Σ_{max}, which may occur if all positions are perfectly matched with the highest possible probability. In this case the score becomes 0 and it can never become negative. C^R_{pred} is a constant which purpose is to increase the contribution to the score of a matching type prediction that has a low reliability R_{pred}. In the current work it was chosen such that $C^R_{pred} + \max R_{pred} = 20$. The reference types $T_{ref}(i)$ are extracted from the reference structure S_{ref} per residue position i as described in section 6.4.

6.5.2 Diversity Measure

As a requirement stated in the problem description, the algorithm should not only find a single very good solution, but rather a number of good solutions as different as possible from each other and from the reference sequence. An effective and simple measure of distance between two sequences is the To obtain a non-negative objective value for minimisation, the average Hamming distance to all other $M-1$ individuals in the current population, minus the sequence length N is computed:

$$F_{div}(A) = N - \frac{1}{M-1}\frac{1}{N} \sum_{\substack{i \in \{1..M\}, \\ A \neq A_i}} d_{Hamm}(A, A_i) \; . \tag{6.3}$$

This function favours individuals farthest away from the rest of the population. In addition, if a sequence similar to the input sequence exists in the population, the function will have a mutually repulsive effect and penalise it. In summary the function addresses two problem requirements: 1) promoting diversity and 2) promoting sequences which are not equal to the reference sequence. Some related works have used the minimum Hamming-distance as diversity objective rather than the average. To investigate the effect of this modification, an alternative definition of Equation 6.3 becomes:

$$F'_{div}(A) = N - \frac{1}{N} \min_{\substack{i \in \{1..M\}, \\ A \neq A_i}} d_{Hamm}(A, A_i) \; . \tag{6.4}$$

The objective value $F'_{div}(A)$ expresses the shortest Hamming distance to any of the $M-1$ other population members. Intuitively this will also degenerate the fitness of individuals which are close to their neighbours in the solution space, but the value depends only on a single other individual, and does not capture the remaining population as a whole.

6.6 Summary

A model of the IFP was presented along with four protein samples each constituting an instance of the problem. The main aim of the model is to make it suitable

for an optimisation algorithms, such as an evolutionary algorithm. By means of secondary structure prediction, a protein sequence can be evaluated with respect to the reference secondary structure. The model is designed for a multiobjectivization approach incorporating the problem requirement of diversity as an objective. Average and minimum Hamming distance to all other individuals in the population are used as variants for the additional objective.

7

Analysing and Mimicking the IFP

7.1 Introduction

Due to the IFPs quickly exploding complexity and highly multi-modal nature, it is a challenging task to determine all or a fraction of its local optima. In addition, tackling real biological instances is computationally expensive which therefore limits the number of possible experiments. In this chapter some of the problem characteristics are identified to design a model that captures the most prominent of these. With a simple definition based on the well-known *NK Model*, the motivation is to make the IFP problem more widely accessible to optimisation algorithm specialists and model experts contrary to being a problem solved mainly by bioinformaticians.

7.2 The NK Model

The *NK Model* introduced by Kaufmann [36] is a tunable rugged fitness function designed to model complex epistatic links among solution loci, to study topics such as gene-interaction. A central feature of the model is its stochastic design which opens up possibilities for statistical analysis of its properties without exact knowledge of all underlying epistatic interactions. While the original model works on a bit-string encoding, Li *et al.* extended the model to continuous and mixed integer solution spaces [42]. Specifically the *nominal discrete NKL model* is of interest, where L denotes the possible values at each locus with $L = 2$ defining the binary case corresponding to the original *NK Model*. The original *NKL Model* is described in Equation 7.1 and shown in Figure 7.1. As implied the value x_i of the ith locus of solution $\mathbf{x}$ and its K neighbouring loci's values x_{i1}, x_{ik} contribute to the final F_{NKL} function value:

$$F_{NKL}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} F_i(x_i; x_{i1}, ..., x_{ik}), \mathbf{x} \in \{0, L\}^N \tag{7.1}$$

Most common neighbourhoods are defined by the K adjacent positions left and right from the position i or K random positions in addition to i, making $K = N - 1$

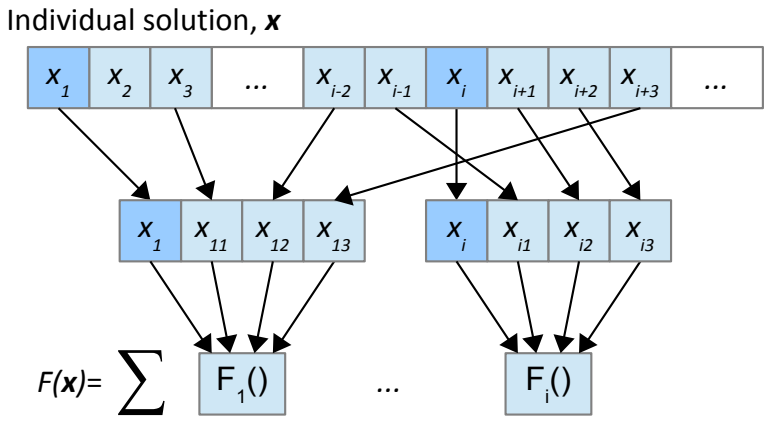


Figure 7.1: The original NKL Model.

the maximum possible value for K. Typically the model is made circular to avoid boundary effects.

7.3 Proposed NKL Benchmark Suite

The proposed *NK model* is presented in Equation 7.2, which is a variation of Equation 7.1. It omits the contribution of the i_{th} position in $\mathbf{x}$, hence K for an identical neighbourhood will be one larger than in the original model and the maximum K becomes $K = N$. This is a minor change that allows to re-create epistatic link effects of the target IFP problem. In addition, the model uses a single function F_0 instead of N different F_i functions. This is for simplicity reasons as N may exceed values of 100. $N = 67$ is chosen because the actual sequence of the target IFP protein *1b3a* has length 67. Then by fixing the number of nominally discrete values possible at each solution position to $L = 20$, a solution vector $\mathbf{x} = \{x_i\},\ x_i \in \{1...20\}$ for the model can be translated 1:1 from an RNA sequence $A = \{aa_i\},\ aa_i \in \{1...20\}$ of the 20 possible amino-acids. This effectively makes the solution encoding of the model and the IFP identical seen from the point of view of an algorithm or solver. Hence, an algorithm designed to work with amino-acid sequences can easily be adapted to solve the proposed model and vice-versa.

$$F(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} F_0(x_{i1},...,x_{ik}), \mathbf{x} \in \{0,L\}^N \tag{7.2}$$

Apart from omitting the contribution of the i_{th} position, the novelty in this work is the combination of two *NK Models*, $F^A(\mathbf{x})$ and $F^B(\mathbf{x})$, with different values of K and different neighbourhood definitions by use of a simple multiplication:

$$F(\mathbf{x}) = F^A(\mathbf{x}) \cdot F^B(\mathbf{x})$$

With this setup, the combined model $F(\mathbf{x})$ can accumulate the characteristics of both its underlying models. Say, strong epistatic interactions are observed between loci i and j in F^A as well as between k and l in F^B. The combined model will then show interactions for *both* pairs i and j as well as k and l. The objective of this setup is ultimately to come as close as possible to the original IFP that features both strong epistatic interactions between close loci, and a constant interaction between loci farther apart.

Two novel *NK Model* instances have been created with the following settings:

- Model 1
  - F^A: a $K = 4$ semi-adjacent circular neighbourhood is designed as follows: $\{x_{i-2}, x_{i-1}, x_{i+1}, x_{i+2}\}$, omitting the central position x_i.
  - F^B: a $K = 3$ neighbourhood of uniform random distribution.
- Model 2
  - F^A: a $K = 4$ neighbourhood as Model 1.
  - F^B: a $K = 5$ neighbourhood of uniform random + 20 positions wide triangular distribution.

The purpose of using a triangle distribution in Model 2 is to induce a higher linkage between loci closer to each other. Essentially the chance of linking two loci drops off linearly until loci are ± 10 positions apart and is then constant. The effects of the presented neighbourhoods used in F^A and F^B on epistatic linkage is seen in Figure 7.10 and discussed further in the following section.

7.4 Sub-sampling the NKL Model

In addition to the neighbourhood variations achievable in the NKL-Model, the need for additional adjustments of solution loci linkage has emerged. The proposed method keeps the number, K, of links and the neighbourhood unchanged to allow for multiple long distance loci interactions, but reduces the strength of these links. In Equation 7.2 the function $F_0(x_{i1}, ..., x_{ik})$ is used to determine the contribution of the solution vector at locus i. The effect of this is that a change of the solution at any of the K positions of the i_{th} neighbourhood basically draws another random number as the new contribution. To reduce the sensitivity to such small perturbations in a solution, a sub-sampling mapping is defined from the L possible values to P where $2 \leq P \leq L$,

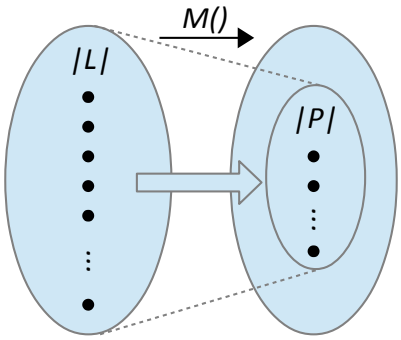


Figure 7.2: Sub-sampling function M().

see Figure 7.2. This means that on average, L/P different values of a given x_{ij} will map to the same lookup value to be passed to F_0. E.g. if $L = 20$ and $P = 5$, any lookup value $1 \leq x_{ij} \leq 20$ will have 4 alternatives yielding the same value $1 \leq M(x_{ij}) \leq 5$. The effect is that more mutations of a solution are required on average to achieve a change in function value than without sub-sampling, because a mutation may hit one of the alternatives that produce no change. Hence the model landscape becomes less rugged (i.e. smoother) while the designed links between loci remain. The modified NKLP model is shown in Figure 7.3.

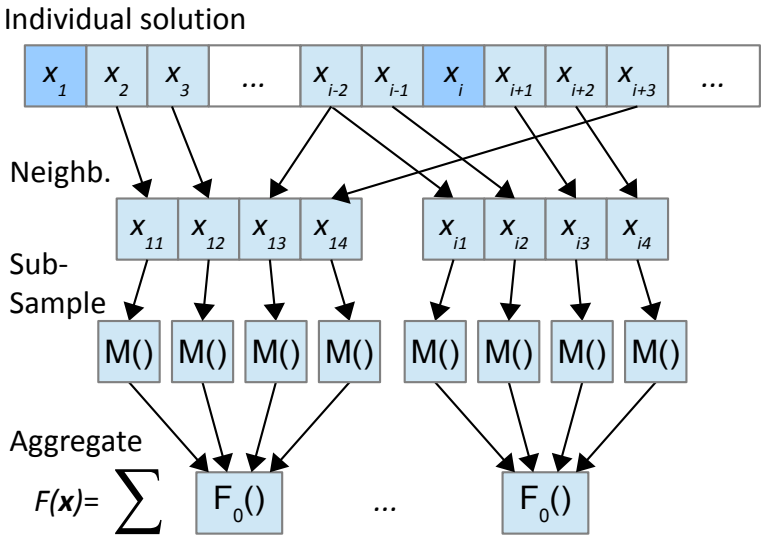


Figure 7.3: The NKLP Model with sub-sampling function M().

7.5 Landscape Analysis

With the introduction of the *NK Model* [36], a number of model features were analysed, mainly by characterising adaptive and random walks in the landscape. Analysis of epistatic links among model variables is another important way of characterising a problem instance, which will be described in the following. As protein sample for the IFP, only *1b3a* is considered as previous work [51] has suggested that different protein samples show very similar characteristics.

7.5.1 Adaptive Walks

An adaptive walk starts at a random position in the objective space and progresses by choosing random *1-mutant* fitter neighbors until no fitter neighbors can be found, and a *local optimum* has been reached. This provides several indicators on the landscape, including the length of such walks and how the number of fitter neighbours decreases with each step.

	Walk length	Fitter, first step	Average fitter	Evaluations	Final Fitness
1B3A	$110.461_{\pm 1.72e+01}$	$592.716_{\pm 1.57e+02}$	$195.932_{\pm 1.79e+02}$	$5601.482_{\pm 3.08e+03}$	$0.165_{\pm 8.85e-02}$
$NK = (67,3)$	$93.290_{\pm 9.66e+00}$	$620.210_{\pm 7.51e+01}$	$167.943_{\pm 1.79e+02}$	$6568.108_{\pm 1.56e+03}$	$-0.896_{\pm 1.10e-02}$
$NK = (67,4)$	$75.030_{\pm 9.60e+00}$	$618.480_{\pm 8.89e+01}$	$164.381_{\pm 1.80e+02}$	$5915.798_{\pm 2.01e+03}$	$-0.869_{\pm 1.00e-02}$
$NK = (67,5)$	$66.000_{\pm 8.50e+00}$	$645.500_{\pm 9.06e+01}$	$161.252_{\pm 1.85e+02}$	$5684.346_{\pm 1.52e+03}$	$-0.850_{\pm 1.20e-02}$
NK1-67	$95.158_{\pm 1.15e+01}$	$625.436_{\pm 8.79e+01}$	$154.981_{\pm 1.77e+02}$	$8042.549_{\pm 3.03e+03}$	$-0.662_{\pm 1.92e-02}$
NK1-67-16	$98.184_{\pm 1.09e+01}$	$635.388_{\pm 9.01e+01}$	$157.682_{\pm 1.79e+02}$	$7542.810_{\pm 2.71e+03}$	$-0.663_{\pm 1.77e-02}$
NK1-67-12	$101.440_{\pm 1.14e+01}$	$653.130_{\pm 8.80e+01}$	$163.296_{\pm 1.87e+02}$	$8240.821_{\pm 2.88e+03}$	$-0.664_{\pm 1.77e-02}$
NK1-67-8	$105.370_{\pm 1.20e+01}$	$646.280_{\pm 9.23e+01}$	$161.384_{\pm 1.82e+02}$	$8600.436_{\pm 3.22e+03}$	$-0.676_{\pm 1.97e-02}$
NK1-67-6	$107.554_{\pm 1.21e+01}$	$641.327_{\pm 8.58e+01}$	$163.479_{\pm 1.81e+02}$	$7672.670_{\pm 2.39e+03}$	$-0.678_{\pm 2.38e-02}$
NK1-67-5	$110.792_{\pm 1.18e+01}$	$638.485_{\pm 8.03e+01}$	$163.726_{\pm 1.79e+02}$	$7590.614_{\pm 2.10e+03}$	$-0.677_{\pm 2.64e-02}$
NK1-67-4	$118.099_{\pm 1.47e+01}$	$638.238_{\pm 8.36e+01}$	$163.196_{\pm 1.79e+02}$	$8436.654_{\pm 2.95e+03}$	$-0.694_{\pm 3.51e-02}$
NK2-67	$81.713_{\pm 9.66e+00}$	$658.267_{\pm 8.86e+01}$	$158.145_{\pm 1.88e+02}$	$7228.492_{\pm 3.02e+03}$	$-0.631_{\pm 1.70e-02}$
NK2-67-16	$82.922_{\pm 9.56e+00}$	$648.660_{\pm 8.79e+01}$	$156.977_{\pm 1.84e+02}$	$7269.606_{\pm 2.51e+03}$	$-0.633_{\pm 1.77e-02}$
NK2-67-12	$85.380_{\pm 8.78e+00}$	$619.530_{\pm 9.06e+01}$	$148.736_{\pm 1.76e+02}$	$7810.186_{\pm 2.96e+03}$	$-0.638_{\pm 1.63e-02}$
NK2-67-8	$89.070_{\pm 1.02e+01}$	$645.110_{\pm 8.92e+01}$	$152.560_{\pm 1.80e+02}$	$7516.714_{\pm 2.81e+03}$	$-0.643_{\pm 1.89e-02}$
NK2-67-6	$91.277_{\pm 1.26e+01}$	$620.327_{\pm 8.07e+01}$	$148.014_{\pm 1.73e+02}$	$7648.270_{\pm 2.74e+03}$	$-0.648_{\pm 2.01e-02}$
NK2-67-5	$93.881_{\pm 1.11e+01}$	$634.455_{\pm 8.07e+01}$	$153.495_{\pm 1.77e+02}$	$7329.120_{\pm 2.39e+03}$	$-0.651_{\pm 2.01e-02}$
NK2-67-4	$98.297_{\pm 1.17e+01}$	$640.703_{\pm 8.71e+01}$	$157.134_{\pm 1.80e+02}$	$7683.431_{\pm 2.44e+03}$	$-0.657_{\pm 2.27e-02}$
1URR	$170.447_{\pm 3.72e+01}$	$900.262_{\pm 3.04e+02}$	$263.579_{\pm 2.50e+02}$	$9138.368_{\pm 4.73e+03}$	$0.144_{\pm 9.31e-02}$
NK1-97	$140.990_{\pm 1.39e+01}$	$917.317_{\pm 1.04e+02}$	$224.573_{\pm 2.58e+02}$	$12906.627_{\pm 4.17e+03}$	$-0.660_{\pm 1.51e-02}$
NK1-97-16	$142.825_{\pm 1.54e+01}$	$925.670_{\pm 1.22e+02}$	$225.925_{\pm 2.60e+02}$	$13161.873_{\pm 4.06e+03}$	$-0.662_{\pm 1.58e-02}$
NK1-97-12	$146.560_{\pm 1.32e+01}$	$927.380_{\pm 1.07e+02}$	$228.028_{\pm 2.59e+02}$	$13103.704_{\pm 3.61e+03}$	$-0.665_{\pm 1.57e-02}$
NK1-97-8	$151.390_{\pm 1.31e+01}$	$914.930_{\pm 9.25e+01}$	$222.447_{\pm 2.54e+02}$	$13559.378_{\pm 3.35e+03}$	$-0.674_{\pm 1.72e-02}$
NK1-97-6	$157.356_{\pm 1.50e+01}$	$919.931_{\pm 9.77e+01}$	$232.213_{\pm 2.56e+02}$	$12945.498_{\pm 3.61e+03}$	$-0.676_{\pm 2.24e-02}$
NK1-97-5	$160.020_{\pm 1.56e+01}$	$935.644_{\pm 8.26e+01}$	$234.587_{\pm 2.57e+02}$	$12236.536_{\pm 3.44e+03}$	$-0.679_{\pm 2.62e-02}$
NK1-97-4	$167.550_{\pm 1.75e+01}$	$924.810_{\pm 8.76e+01}$	$233.023_{\pm 2.53e+02}$	$12960.835_{\pm 3.73e+03}$	$-0.685_{\pm 3.80e-02}$
NK2-97	$117.495_{\pm 1.15e+01}$	$919.277_{\pm 1.01e+02}$	$215.164_{\pm 2.55e+02}$	$12581.155_{\pm 3.82e+03}$	$-0.633_{\pm 1.59e-02}$
NK2-97-16	$118.864_{\pm 1.16e+01}$	$923.078_{\pm 1.16e+02}$	$215.512_{\pm 2.56e+02}$	$12189.403_{\pm 4.47e+03}$	$-0.635_{\pm 1.43e-02}$
NK2-97-12	$124.860_{\pm 1.18e+01}$	$928.520_{\pm 9.80e+01}$	$212.186_{\pm 2.55e+02}$	$13597.801_{\pm 5.22e+03}$	$-0.643_{\pm 1.43e-02}$
NK2-97-8	$127.010_{\pm 1.33e+01}$	$929.160_{\pm 1.12e+02}$	$220.158_{\pm 2.58e+02}$	$12875.790_{\pm 4.20e+03}$	$-0.641_{\pm 1.54e-02}$
NK2-97-6	$131.455_{\pm 1.28e+01}$	$913.683_{\pm 1.19e+02}$	$217.635_{\pm 2.54e+02}$	$13076.175_{\pm 3.76e+03}$	$-0.649_{\pm 1.44e-02}$
NK2-97-5	$135.505_{\pm 1.34e+01}$	$916.030_{\pm 1.04e+02}$	$220.599_{\pm 2.53e+02}$	$12621.200_{\pm 3.86e+03}$	$-0.650_{\pm 1.76e-02}$
NK2-97-4	$145.970_{\pm 1.33e+01}$	$930.830_{\pm 1.01e+02}$	$218.461_{\pm 2.53e+02}$	$12652.210_{\pm 3.93e+03}$	$-0.659_{\pm 1.81e-02}$

Table 7.1: Adaptive walks statistics

From literature it is known that the length of an adaptive walk on a *NK Landscape* will decrease for larger K values regardless of the choice of neighbourhood. This is due to the induced ruggedness when using larger K. The effect can be seen in Table 7.1 where the *Model 2* instances have shorter walks than *Model 1* instances for the otherwise same settings. The table is organised in two main parts; the top part concerns the 1B3A IFP sample of length $N = 67$, the lower part concerns 1URR with length $N = 97$. Alongside the IFP instances are their *NK Model* counterparts of same length and different sub-sampling values, P. Concentrating on the 1B3A IFP sample, it can be seen that the effect of combining two *NK Models* increases the length of a walk approaching that of the IFP for *Model 1*. All models show almost the same average number of fitter neighbours at the first step, ± 636.5, which is exactly half of the neighbourhood size of $N \cdot (L-1) = 67 \cdot 19 = 1273$. This number shows higher variation in the *IFP* problem, indicating more location-dependent characteristics than those expressed in the *NKL* models. The number of evaluations required on average to reach a local optimum is a bit higher for the *NK Models* without sub-sampling, but with sub-sampling, $P = 5$ the adaptive walk length can be matched to that of the IFP: $110.461_{\pm 1.72e+01}$ vs. $110.792_{\pm 1.18e+01}$. The *Model 2* however cannot reach those lengths even with $P = 4$ yielding 98.297, though the magnitude of the difference is a fraction of the number. Turning to the second IFP sample, 1URR, it is noticeable that the increased length of $N = 97$ also increases the lengths of adaptive walks of both *NK Models*. Meanwhile the IFP also shows a longer walk length, and with $P = 4$ *Model 1* with length $167.550_{\pm 1.75e+01}$ can barely reach the same length as the IFPs $170.447_{\pm 3.72e+01}$. Again the *Model 2* shows a shorter length for the lowest sub-sampling $P = 4$, in this case 145.970. All in all the *NKL Model* statistics can roughly be fitted within a maximum factor of two of the *IFP* problem, and in most cases a far better match is achieved.

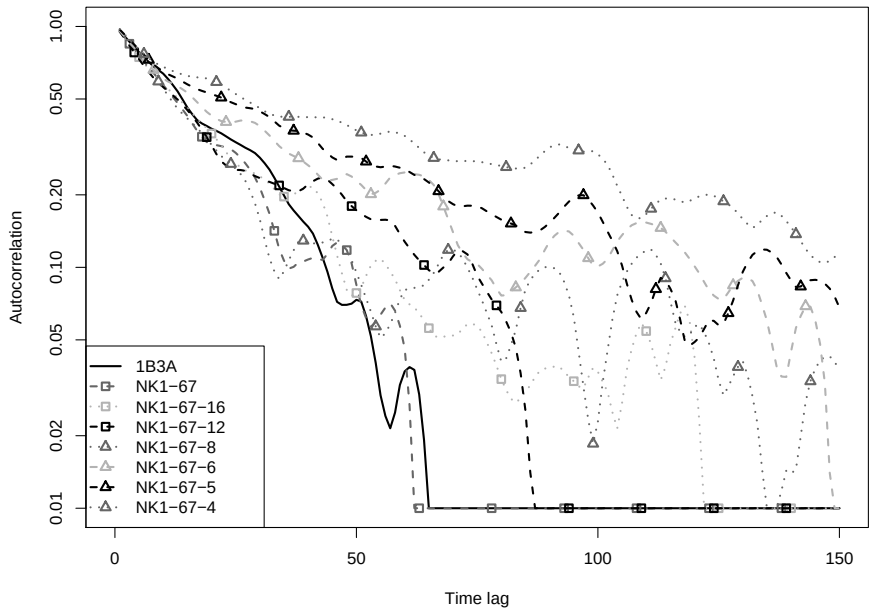


Figure 7.4: Random walk analysis of 1B3A vs. NK1 variants

7.5.2 Random Walks

To compute the auto-correlation function of the problem and models, a number of random walks have been performed starting from a local optimum. The reason for choosing a local optimum as a starting point is motivated by the fact that the main dynamics of the estimated secondary similarity score F_{sec} are present only when the predicted structure matches the reference structure well.

Evolutionary algorithms will mostly be evolving around such good solutions, and small perturbations in key positions here have larger impact on the overall match score than in a random poor matching solution. The auto-correlation function used in this work is equivalent to the one in [35] and can be written as co-variance of function values at t and $t+s$ over the product of their deviations.

$$R(t,s) = \frac{\sigma(F(\mathbf{x}_t, F(\mathbf{x}_{t+s}))}{\sigma(F(\mathbf{x}_t)) \cdot \sigma(F(\mathbf{x}_{t+s}))} \tag{7.3}$$

As the walks all start from local optima, the ideal analysis would use $t = 1$, and analyse decay in correlation as the hamming-distance s increases. As the random walks were repeated 100 times from different local optima for the IFP the resulting auto-correlation progress is somewhat noisy. Therefore the analysis is extended to $t \in \{1..10\}$ to include more information, yet still analysing the landscape relatively

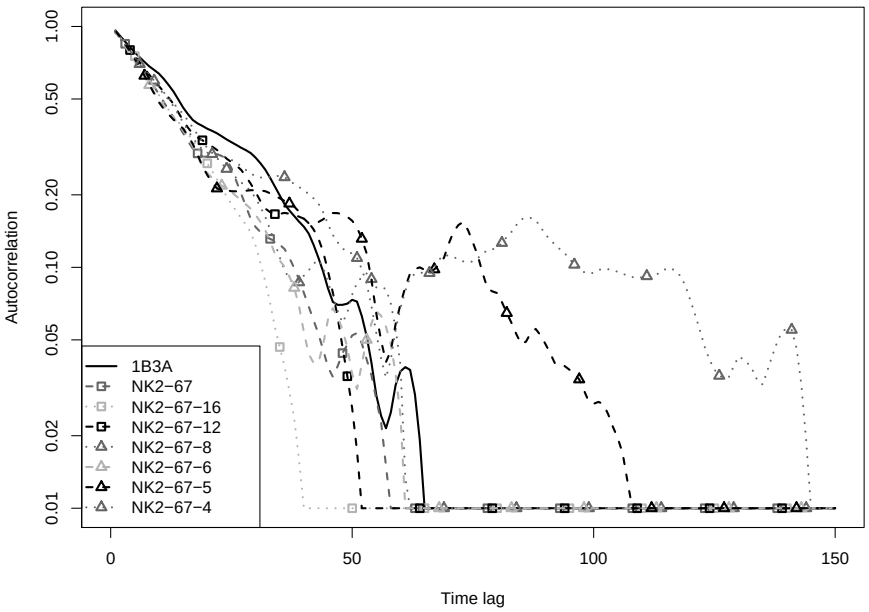


Figure 7.5: Random walk analysis of 1B3A vs. NK2 variants

close to local optima. Figures 7.4 to 7.7 show the resulting decay in autocorrelation for IFP instances 1B3A ($N = 67$) and 1URR ($N = 97$) for both *NK Models*. A general observation is that reducing the sub-sampling values, P, increases the correlation length as expected. This can be explained by the loosened epistatic links which make the model less sensitive to small changes, which in turn means that a solution has to be perturbed more to express the same change in fitness value.

For the shorter protein sample 1B3A, the original $N = 67$ models without sub-sampling i.e., $P = 20$, produce quite similar correlation lengths as the IFP of around 60. The *Model 2* variants show slightly shorter lengths than *Model 1* variants in general, and also for the slope of the decay, Model 2 is in general steeper than the Model 1, but this effect can be efficiently tuned to adapt to the target. For example, variant NK2-67-12 has similar correlation length, but also slope on Figure 7.4. The longer protein sample 1URR shows a correlation length of over 300 and to capture this characteristic a lot of sub-sampling is required, i.e. with a setting of P around 5 being optimal.

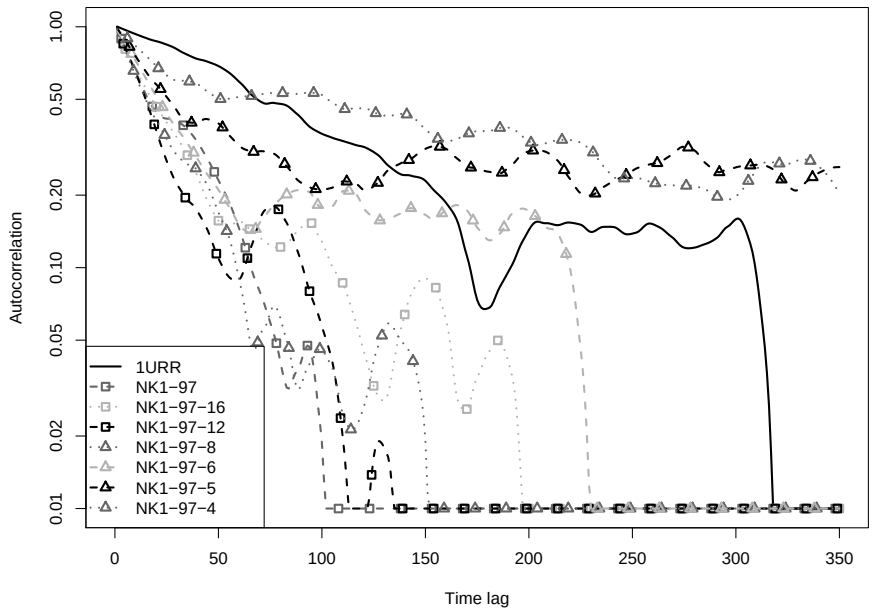


Figure 7.6: Random walk analysis of 1URR vs. NK1 variants

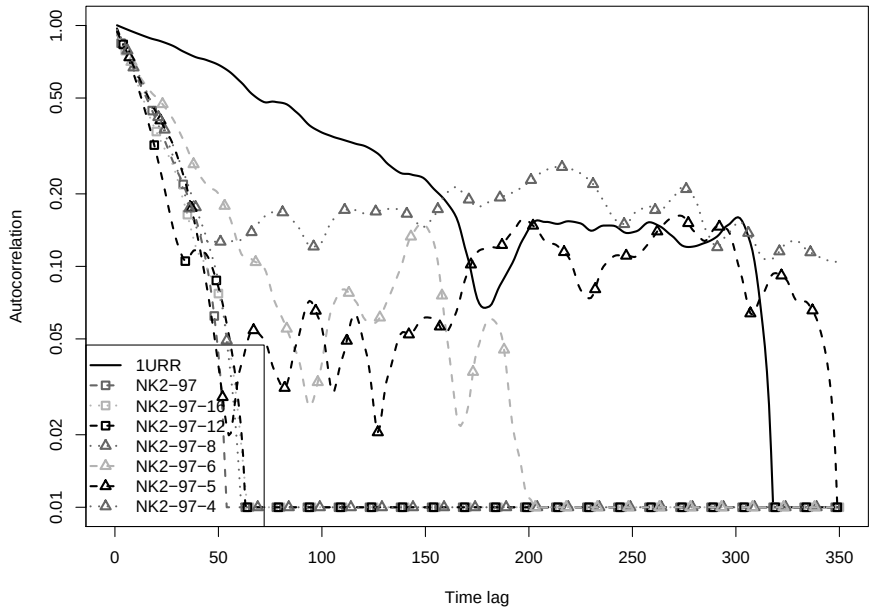


Figure 7.7: Random walk analysis of 1URR vs. NK2 variants

7.5.3 Epistatic Link Analysis

Epistatic interaction is a concept borrowed from genetics where a gene can be defined as being epistatically linked if its effect depends on the state of other genes. In this analysis of epistatic links, loci of a solution are examined pairwise in a systematic manner. To fully discover all such links would require to observe the effect of all possible variable combinations of two loci in all possible states of the rest of the solution vector. As this would require too many function evaluations, a local optimum is again chosen as the starting point, keeping all variable loci constant except for the pair being sampled. For a pair of selected loci i and j, $i \neq j$, three additional function evaluations are done evaluating first a mutation at i, then a mutation at j computing the error $\varepsilon(\mathbf{x}, i, j)$ by comparing to the same mutations at both i and j at the same time:

$$\varepsilon(\mathbf{x}, i, j) = |\Delta F(\mathbf{x}_{(i,j)}) - (\Delta F(\mathbf{x}_{(i)}) + \Delta F(\mathbf{x}_{(j)}))|$$

Where $\Delta F(\mathbf{x}_{(\mathbf{y})})$ denotes the function value difference in F when the solution $\mathbf{x}$ has values substituted at loci $\mathbf{y}$. If there is no linkage between locus i and j at local optimum $\mathbf{x}$, $\varepsilon(\mathbf{x}, i, j)$ will be *zero* for all possible substitution pairs. This information is typically expressed on matrix form, but reduced here to a single vector, averaging the linkage in terms of loci distance $d = |i - j|, i \neq j$. Figure 7.9(a) and (b) show

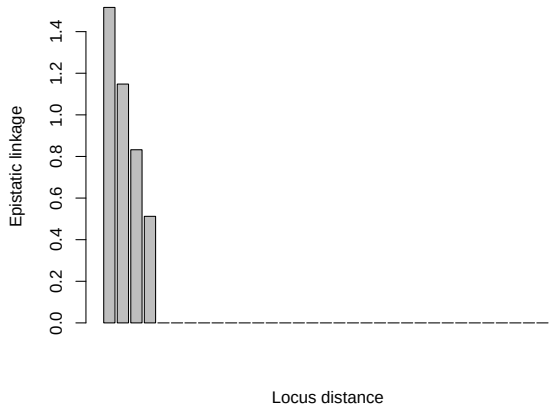
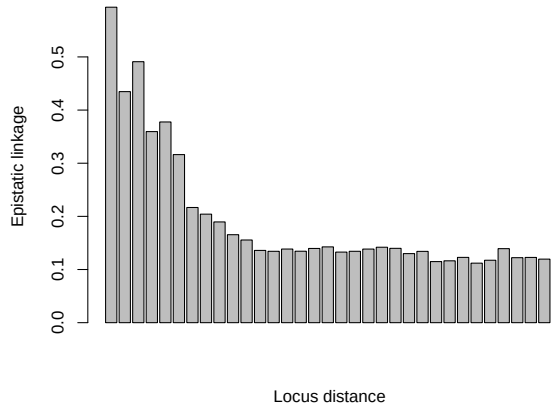


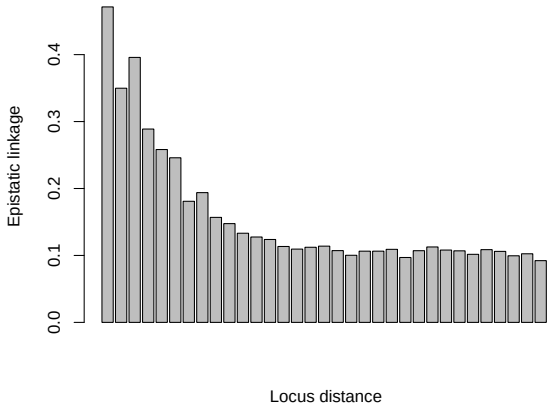
Figure 7.8: Epistatic linkage in local optima of the NKL Model.

this epistatic linkage at two different random local optima of the *IFP* problem. Figure 7.10(a) and (b) shows linkage of the proposed models at a local optimum and Figure 7.8 the standard *NKL Model* with $K = 5$ for comparison.

Clearly the standard *NKL Model* has absolutely no linkage beyond 5 loci apart, which is achieved in the combined models proposed here with the second function F^B having almost uniformly distributed neighborhood. To achieve the ramp down which



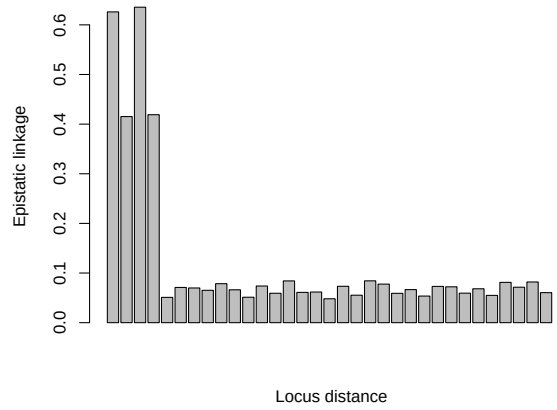
(a) Local optimum 1



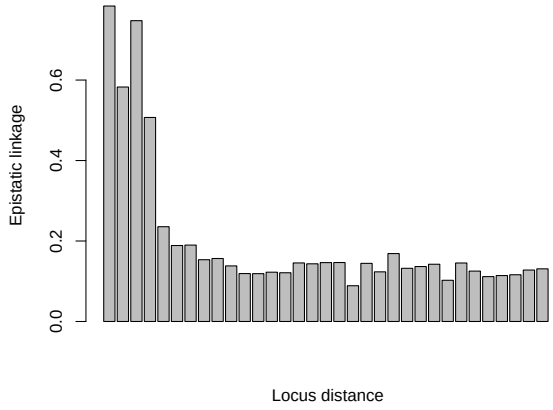
(b) Local optimum 2

Figure 7.9: Epistatic linkage in two local optima of protein *1b3a*.

can be observed in the real *IFP* problem, the neighborhood of function F^B of *Model 2* is generated from a partially triangular distribution, which effect is quite noticeable in Figure 7.10(b). The epistatic links are slightly stronger between close loci in the models than in the IFP but long range interactions look very similar in both models. The other main feature of the real *IFP* problem is the characteristic dip and then rise in locations 2 and 3 apart which is captured by the neighborhood function of F^A and observed in both combined models in Figure 7.10(a) and (b).



(a) Model 1



(b) Model 2

Figure 7.10: Epistatic linkage in local optima of proposed models.

7.6 Conclusions

This chapter introduced a benchmark problem based on the well know *NK Model* extended to a *nominal discrete NKL Model* definition in a previous third-party work. Setting $L = 20$ allows the model to work with amino-acid like sequences similar to RNA with the ultimate goal of mimicking the Inverse Folding Problem (IFP). With the novel sub-sampling parameter P additional model characteristics can be fine-tuned to match the target IFP problem. With the same amount of epistatic links, their strength can be reduced overall to make the model less sensitive to small perturba-

tions in solutions. Thorough problem analysis was conducted through adaptive- and random walks in terms features like fitter neighbors, auto-correlation among others as well as an extended epistatic linkage sampling around local optima. Very similar characteristics within an upper bound of a factor two were achieved in almost all tests when comparing the *NKL Model* instances to the IFP. Running selected Genetic Algorithms with different diversity maintaining features in the following should also show very similar convergence behavior in diversity and fitness for the proposed models and the IFP. Furthermore the statistical nature of the *NK Model* with existing proofs and lemmas may provide the ground for a theoretical estimate on the number of protein sequences which fold into a given protein structure.

8

Designing Diversity Preserving Algorithms

8.1 NSGA-II with Diversity as Objective and Quantile Constraint

With the aim to address the requirements stated in the problem description an algorithm employing multiobjectivization is presented and described in this chapter. Turning the requirement of finding many different solutions into an additional objective also has the advantage of increasing the diversity of the population which emphasises exploratory behaviour of the algorithm. This is favourable feature for solving *multi-modal* problems with many wide-spread globally good as well as deceptive solutions. The design is based on the well known Non-dominated Sorting Genetic Algorithm (NSGA-II) [16], a multi-objective genetic algorithm with a simple but effective dominance sorting mechanism. In addition to the additional diversity objective, $F_{div}(A)$, two crucial modifications of the original algorithm were done. These are highlighted in Algorithm 3: removal of doubles described in section 8.1.2 and Quantile Constraint (QC) in section 8.1.3. The QC mechanism has a single setting which efficiently influences the exploratory vs. exploitative behaviour of the algorithm, which turned out to be a necessity for best performance in both aspects.

8.1.1 Non-dominated Sorting

As discussed in section Section 3.5.3 directly determining the better individual in the multi-objective case can be difficult. With reference to Fig. 3.4 describing the *Pareto front*, the terminology is extended with the *Pareto rank*, a similar concept depicted in Fig. 8.1. The 2_{nd} *Pareto rank* can be determined by first excluding the *Pareto front*, or 1_{st} rank, then compute the *Pareto front* of the remaining points.

With the described procedure it is not possible to distinguish all individuals in terms of fitness, but they can all be assigned a rank. The rank enables to separate the individuals in sets where all members of a lower *Pareto rank* set can be said to

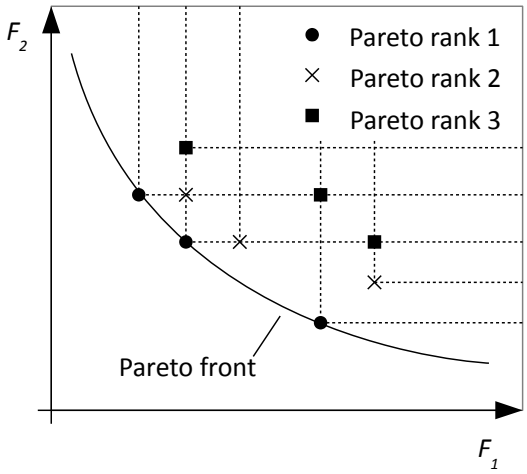


Figure 8.1: Pareto ranks in objective space of objective functions f_1 and f_2 in minimisation.

be better than a higher ranked set. In Algorithm 3 this sorting method is used before truncating the $2N$ sized union of parent and offspring populations $P_t \cup Q_t$ down to the size N. Obviously it is preferred to preserve all individuals in the lower rank sets as well as the population *Pareto front*.

8.1.2 Removal of Doubles

In the context of diversity preservation and due to the problem statement, having two or more identical individuals in the population is undesired. Especially as in [64] when diversity for a sequence A is defined as the minimal distance to any other sequence A', a sequence $A = A'$ must be avoided. With the diversity calculation proposed in Section 6.5.2, this issue has less impact, but nevertheless doubles are removed in this work. The procedure is executed in line 6 of Algorithm 3 after the application of genetic operations and before non-dominated sorting and crowding based truncation of the unified population R_t takes place in NSGA-II.

When two identical sequences are detected, one of them is mutated with a probability of $\frac{5}{N}$ to distance the individual with a Hamming-distance of 5 on average.

8.1.3 Quantile Constraint

Clearly it is easy to generate individuals which contribute to diversity, as any random individual will achieve this. On the other hand individuals with good fitness in terms of $F_{sec}(A)$ are much harder to generate and should therefore receive higher priority over individuals with good fitness in terms of $F_{div}(A)$. To address this imbalance, the Quantile Constraint (QC) was introduced at the end of every generation, in line 9 of Algorithm 3. Given a quantile size C_q, the population P_t at time t is divided according

Algorithm 3 DAO-QC NSGA-II

1: $Initialise(P_0)$ {randomly generated individuals}
2: $t \leftarrow 0$
3: **while** $t < t_{max}$ **do**
4: $Q_t \leftarrow makeNewPop(P_t)$ {selection, mutation, re-combination}
5: $R_t \leftarrow P_t \cup Q_t$
6: $mutateDoubles(R_t)$ {eliminate doubles by mutation}
7: $F \leftarrow fastNonDominatedSort(R_t)$
8: $P_t \leftarrow truncate(F)$ {based on domination and crowding}
9: $setQuantileConstraint(P_t)$ {to penalise worst quantile}
10: **end while**

to $F_{sec}(A)$ into a $C_q\%$ sized partition and a $100 - C_q\%$ sized partition. All individuals in the former, less fit, partition are assigned a constraint penalty that prevents the constrained individuals from mating and surviving the next generations. Hence the population is cleaned from individuals far spread in the solution space, but with poor $F_{sec}(A)$ score. The selection pressure can then be selectively adjusted by changing the size of the quantile C_q. Preliminary test have shown that a value around $10 - 25\%$ represents the best trade-off in most cases.

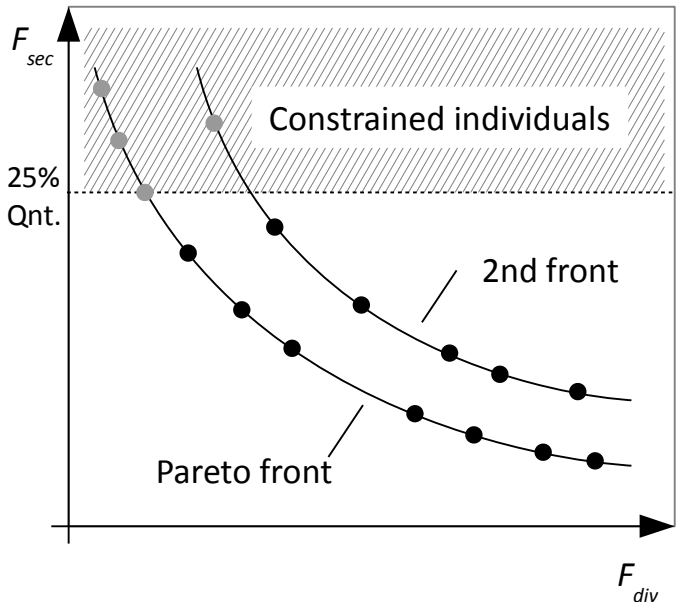


Figure 8.2: Quantile constraint approach with $QC = 30\%$. Minimisation problem shown, with F_{sec} being the main objective, F_{div} designed for diversity preservation.

8.2 Preference Based Genetic Algorithm

In this section another algorithm design is presented which has as a goal to directly maintain a population that respects a predefined user preference. Just as for the NSGA-II DAO-QC presented previously in Section 8.1 it is desired to emphasise diversity in the final algorithm population as it was stated by the problem description along with the primary fitness function. The amount of diversity induced will directly affect the balance between the algorithms exploration and exploitation characteristics which is crucial for the performance of the algorithm. It is therefore parametrised by the user through a pre-set preference settings used by the algorithm to determine the set of solutions best satisfying the preferences each generation of the algorithm. One significant motivation and advantage of the approach is that the preference mechanism allows to shape the population characteristics directly in place according to the preferences. This mindset opposes other approaches where certain characteristics are sought after by modifying algorithm operators or mechanisms which as a side-effect induce the desired effect to a degree after a certain number of generations.

8.2.1 Population Preference

In this first version of the Preference Based Evolutionary Algorithm (PBEA), preference is described by means of a Weighted Sum Model (WSM). The WSM is a well-known and simple multi-criteria decision making method for evaluating a number of alternatives in terms of a number of decision criteria. In this case the criteria are the two preference terms *average population diversity* and *average population fitness* and by tuning the weights the algorithm user can shift focus arbitrarily between one or the other. The weighted sum score of a given population P is calculated as follows:

$$F_{pref}(P) = -W_{fit} \cdot F_{fit}(P) + W_{div} \cdot F_{div}(P) \tag{8.1}$$

Where $F_{fit}(P)$ and $F_{div}(P)$ are population *average fitness* and *diversity*. The respective weights are W_{fit} and W_{div} where the former is negated due to the fact that diversity is to be maximised but fitness minimised.

8.2.2 Process Description

With the goal of constantly maintaining a current population best fulfilling the defined preferences, the weakest individuals from the $2N$ sized combination of parent and offspring populations are determined and removed until the desired population size of N is achieved. Algorithm 4 below shows the pseudo-code for the PBGA. As the preference score of the population depends on the contribution of all individuals one cannot assign a fix score or fitness to each individual. One can, however, always determine the immediate contribution of an individual in the current population by removing it and comparing the preference score of the remaining population with

that of the population including that individual. The procedure *getWeakestIndividual* of determining the weakest individual in Algorithm 4, step 7 is defined as follows:

1. Systematically remove one individual
2. Compute the weighted sum score according to Equation 8.1
3. Add the individual back to the population
4. Repeat from step 1. until all individuals have been tried once and the worst individual can be determined.

With this approach the weakest individual may be determined and removed. The process must then start over according to step 6, as at this point the population has changed and all individual contributions will likewise have changed slightly. Some care needs to be taken when implementing the truncating of the population from $2N$ to N individuals, especially if the population size N is large. The order of complexity is polynomial and can quickly become a significant factor especially when the evaluation of the primary objective function is relatively short.

Algorithm 4 Preference-Based Genetic Algorithm

1: $Initialise(P_0)$
2: $t \leftarrow 0$
3: **while** $t < t_{max}$ **do**
4: $Q_t \leftarrow makeNewOffspringPop(P_t)$
5: $R_t \leftarrow P_t + Q_t$
6: **while** $|R_t| > |P_t|$ **do**
7: $I \leftarrow getWeakestIndividual(R_t)$
8: $R_t \leftarrow R_t - I$
9: **end while**
10: $P_t \leftarrow R_t$
11: $t \leftarrow t + 1$
12: **end while**

8.3 Summary

In this chapter, two new algorithms were presented both having the goal of optimising fitness and diversity in the population simultaneously. The NSGA-II DAO-QC achieves this by adding an additional objective designed to incorporate the individuals distance to other individuals in the population - either average or minimum distance has been implemented. With the PBGA an even more direct approach has been developed to evaluate the population as a whole. The combination of individuals that fulfil the global criteria best is found by iteratively identifying and removing the individual contributing the least in combined parent and offspring population. Both algorithms

have built-in mechanisms to control the emphasis on diversity vs. fitness which is closely, if not directly, related to exploration vs. exploitation charactersitics of the algorithms. The NSGA-II DAO-QC achieves this by applying a Quantile Constraint (QC) to a predefined quantile of the population to eliminate the worst individuals in terms of fitness. In the PBGA, the preference is expressed by means of a Weighted Sum Model (WSM) with two terms and their corresponding weights. The WSM is then able to evaluate how well the population as a whole expresses the user preference w.r.t. exploration vs. exploitation.

Part III

Algorithm Application and Experiments

9

Experiments Planning and Setup

9.1 Introduction

At this point a number of algorithms have been presented, all with common and specific settings. In addition, two benchmark problems with their own settings were introduced as well as the four real protein instances. This all adds up to a large number of possible setup combinations for the experiments, and some structuring and selection is a must: In general, four individual sub-studies are conducted for the real IFP model instances in Chapter 10 and for the NK model instances in Chapter 11. These four studies are:

- *Baseline Study* - compares selected existing genetic algorithms
- *Diversity as Objective with Quantile Constraint Study* - compares algorithm settings
- *Preference Based Genetic Algorithm study* - compares algorithm settings
- *Summary of Algorithm Experiments study* - compares DAO-QC NSGA-II to PBGA

In a study, the algorithm performance is evaluated on each problem instance based on diversity and fitness in the algorithms population averaged over 30 individual runs. Through convergence plots, the progress of these values is analysed and in a cross comparison the final values are compared. The Wilcoxon test indicator [72] with a 5% significance level provides statistical confidence in comparing the sets with symbols '▲', '▽' and '-' indicating superior, inferior and no significant difference in the algorithm performance. For fitness, smaller values are desirable while larger diversity values are regarded as better. Therefore the orientation of the triangle is adapted to express the aim of the value being statistically tested and the symbol '▲' will always imply the preferred values. Some of the relevant experiments not shown in Chapters 10 and 11 can be found in the Appendix where the exact same structure of four studies is followed.

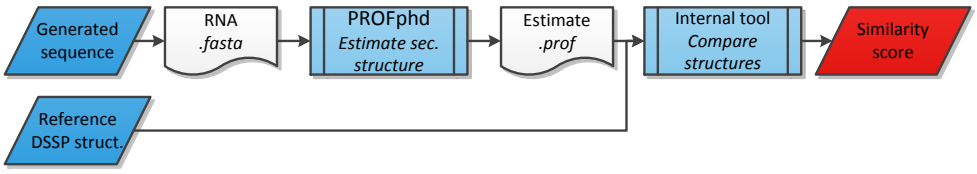


Figure 9.1: Process flow for prof secondary structure similarity evaluation

9.2 Secondary Structure Prediction Configuration

Figure 9.1 shows the work-flow of secondary structure scoring with input data to the left, processing by prof and output parsed and converted to a score. The secondary structure prediction tool used is a further development of the tool PROFphd [59], named ReProf. ReProf, like PROFphd, uses a pre-trained neural network and is said to have a success-rate of above 70%. The tool takes as input a *.fasta* file with the amino-acid sequence and returns a tabular prediction report with one row per residue position. The columns of interest are *PHEL* and *RI_S* which contain "PROF predicted secondary structure: H=helix, E=extended (sheet), blank=other (loop)" and "reliability index for PROFsec prediction (0=lo 9=high)". For each protein sample another secondary structure annotation is required to match with the prediction. The "Define Secondary Structure of Proteins" (DSSP) tool [34] can generate an annotation report from tertiary structure information. As only the three structure types, *Helices (H)*, *Sheets (E)* and *Loops (L)* are considered by the optimisation model, some simplification is required. In the documentation of DSSP, the following possible annotation types are found:

- G = 3-turn helix (310 helix). Min length 3 residues.
- H = 4-turn helix (α helix). Min length 4 residues.
- I = 5-turn helix (π helix). Min length 5 residues.
- T = hydrogen bonded turn (3, 4 or 5 turn)
- E = extended strand in parallel and/or anti-parallel β-sheet conformation. Min length 2 residues.
- B = residue in isolated β-bridge (single pair β-sheet hydrogen bond formation)
- S = bend (the only non-hydrogen-bond based assignment).
- C = coil (residues which are not in any of the above conformations).

With *Helices* characterised by a corkscrew shape, *sheets* as parallel connected segments, and *loops* as everything else, the above structure types are simplified as follows:

$$G,H,I \Rightarrow H;\ E,B \Rightarrow E;\ T,C,S \Rightarrow L$$

Secondary structure predictions and reliability are noted $T_{pred}(i)$ and $R_{pred}(i)$ respectively for each position, i. They are matched against a reference *.dssp* file with secondary structure annotations, $T_{ref}(i)$ to produce the similarity score. The

matching is part of the model described by Equation 6.2 in Section 6.5.1.

9.3 Common Settings

To minimise the influence of specific algorithm settings they all share the same parameters as far as possible. Obviously all algorithms receive the same evaluation budget of 90000 evaluations to allow a fair comparison. Also a population size of 100 individuals is kept the same, to evaluate the diversity of the found individuals on the same basis. All algorithms require *selection*, *crossover* and *mutation* operators and these were chosen as the standard *Binary Tournament*, *1-point Crossover* with probability 0.9 and *Uniform Mutation* with probability $\frac{1}{N}$ respectively. These settings are summarised in Table 9.1.

Table 9.1: Common algorithm settings

Setting	**Value**
Population size	100
Termination condition	90000 function evaluations
Selection	Binary tournament (BT)
Crossover operator	1-point, p_c=0.9
Mutation operator	Uniform, $p_m = \frac{1}{N}$
Duplicate individuals	No action / Uniform mutation, $p_m = \frac{5}{N}$
IFP problem instances	1B3A, 256B, 1OAI, 1URR
NK-Model instances	$Model \in \{1,2\}$, $N =\in \{67,97\}$, $P =\in \{4,20\}$)

9.4 Baseline Study

For baseline comparison, common algorithms were tested on the real and benchmark problems with the purpose of providing a reference for further assessments of the of the proposed algorithms. The settings used are listed in Table 9.2, and are mostly standard settings. The generational GA (gGA) was is detailed in Algorithm 1, the steady state GA (ssGA) is a simpler version where each generation consists of mating only two selected parents and inserting them in the population by means of a replacement strategy. The synchronous cellular GA (scGA) was discussed in Section 5.4.5. To obtain an algorithm with higher focus on diversity, the generation GA with doubles removal (rdGA) has been added here and is also selected to be included in all following experiments for reference comparison. It uses the same doubles mutation strategy as the DAO-QC NSGA-II described in Section 8.1.2 which will allow to isolate the effect of the doubles removal mechanism. Additional arguments for the choice are that it is the better diversity preserving method among the four, that also is more successful in finding the best solution.

Table 9.2: Algorithm settings, baseline study

Setting	Value
Algorithm	rdGA, gGA, scGA and ssGA
Selection	Binary tournament
Crossover operator	1-point, $p_c = 0.9$
Mutation operator	Uniform, $p_m = \frac{1}{N}$
Neighbourhood	C9 (scGA)
Elitism	2 individuals (gGA)
Replacement strategy	replace worst (ssGA)
Common settings	

9.5 Diversity as Objective with Quantile Constraint Study

The second set of experiments are designed to asses the performance and influence of the Diversity as Objective (DAO) with Quantile Constraint (QC) approach. Table 9.3 summarises the settings specific to the DAO-QC study experiments. Three different values of the quantile constraint C_q are considered for DAO-QC NSGA-II: 0%, 10% and 25% of the population. The diversity objective has been tested both using average and minimum Hamming distance variants, see Equation 6.3 and 6.1. In the experiments DAOA and DAOM denotes average and minimum Hamming respectively. As targets are chosen the four protein instances and the NKL instances. These initial experiments on one hand focus on analysing the effect of different quantile constraint settings on the proposed algorithms' performance. On the other seek to demonstrate the similarities between the IFP and the NKLP model. For this reason the second set of experiments is expanded by testing multiple values of the sub-sampling value P.

Table 9.3: Algorithm settings, DAO with QC study

Setting	Value
Algorithm	rdGA, NSGA-II DAO-QC
Quantile constraint	$C_q \in \{0\%, 10\%, 25\%\}$
Diversity Objective	$\{MeanHamming, MinHamming\}$
+ *Common settings*	

9.6 Preference Based Genetic Algorithm Study

The PBGA was tested with the following six weight ratio settings intended to express a variety of behaviours from low to high diversity:

$$W_{(fit,div)} = \{(1.0,0.0),(0.9,0.1),(0.8,0.2),(0.7,0.3),(0.5,0.5),(0.3,0.7)\}.$$

For each 2-tuple the first element corresponds to the fitness weight W_{fit} and the second element corresponds to the diversity weight W_{div}. Table 9.4 summarises the settings and parameters used to conduct the experiments.

Table 9.4: Algorithm settings, PBGA study

Setting	Value
Algorithm	rdGA, PBGA
Weights	$\{(1.0,0.0),(0.9,0.1),(0.8,0.2),(0.7,0.3),(0.5,0.5),(0.3,0.7)\}(PBGA)$
+ *Common settings*	

9.7 Summary of Algorithm Experiments Study

To compare convergence between the two developed algorithms, this study combines results of them both, side-by-side. From previous runs some of the most efficient settings have been identified and will be used.

Table 9.5: Algorithm settings, algorithm summary study

Setting	Value
Algorithm	rdGA, NSGA-II DAO-QC, PBGA
Quantile constraint	$C_q \in \{10\%, 25\%\}$ (DAO-QC)
Diversity Objective	$\{MeanHamming, MinHamming\}$ (DAO-QC)
Weights	$\{(0.9,0.1),(0.8,0.2)\}(PBGA)$
+ *Common settings*	

10

Experiments on the IFP

10.1 Introduction

In this chapter the various algorithms will be tested on the real protein samples following the setups described in Chapter 9. Performance is presented both in terms of final values and convergence to identify the optimal settings for different problem instances and algorithms.

10.2 Baseline Study

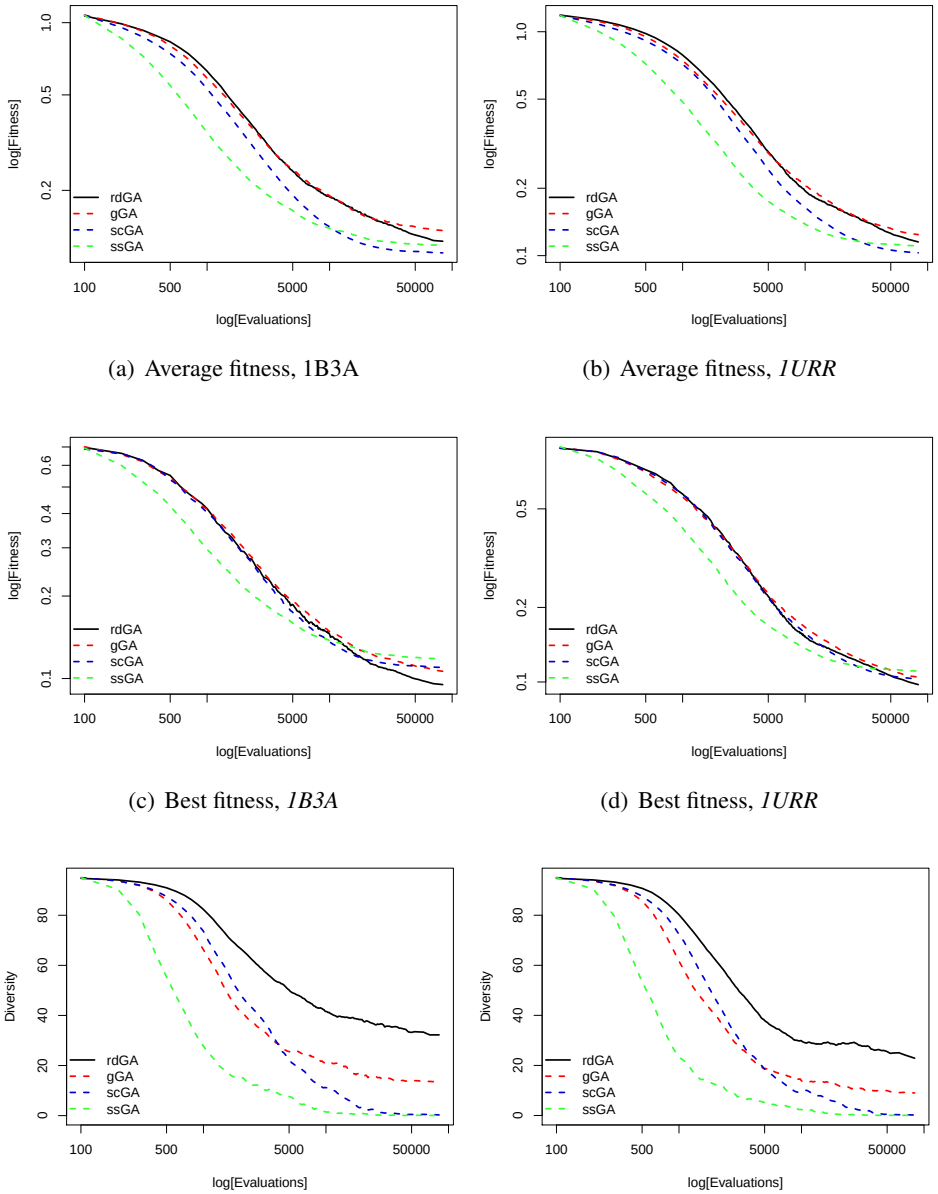
As stated in Chapter 9, the purpose of the baseline study is to get an initial view of the general performance of existing algorithms. The convergence plots concentrate in this section on the IFP instances corresponding to the protein 1B3A and 1URR, but the equivalent experiments on the two remaining proteins can be found in the Appendix in Section A.2. In Figure 10.1(a-f) are shown aggregated convergence plots of average fitness, best fitness and average diversity side-by-side for the two proteins. The general picture is overall the same with a few exceptions. A bend in the curve at $5-8000$ evaluations marks a sort of transitioning and suggests that the rdGA converges a bit later with the longer 1URR. The effect of removing doubles can be regarded as a continuous game of lotto, with a little, but effective chance of discovering new solutions as seen by the constant final logarithmic slope up to 90000 evaluations. The two periods separated by the $5-8000$ evaluation mark are also visible in the diversity plots, where the rdGA maintains an almost constant diversity in the second period. The effect lets the rdGA outperform the other algorithms in terms of best fitness, but not average fitness, which is explained by the elevated diversity. In an extended time frame it is likely that the rdGA would catch up on average fitness as well. The cellular GA, scGA clearly has diversity preserving characteristics until around $5-10000$ evaluations, with almost zero diversity above 40000 for 1B3A and 50000 for 1URR. Though at this point the scGA has converged it has manages to find a good solution superior to the steady state ssGA on average.

Final values at 90000 evaluations for the four samples are captured in Table 10.1. Best and worst series value of the experiment is marked with dark and light grey respectively. The Wilcoxon test indicator [72] with a 5% significance level provides statistical confidence in comparing the best and worst valued series only (i.e., the other series are not tested). For the triangle symbols '▲' and '▽' to be added, the series has to be significantly different statistically from all the others in the experiment. If for one or more series this is not the case, the '-' symbol is noted. The representation regards only The complete source data for the tables and plots is available in Appendix A.

The rdGA always finds the best individual in all experiments with statistical significance. In terms of average fitness the scGA is always better, but it is not indicative as the population has zero diversity making it equivalent to the best fitness value. Also in terms of diversity, the rdGA achieves much higher values between 20% and 30%. Noticeable is the lower diversity on the longer samples. This can possibly be explained by the fact that the population size of 100 limits the extend of the search space that a GA can explore meaningfully.

		Average fitness	Best fitness	Average diversity
1B3A	rdGA	$0.122_{\pm 1.14e-02}$	$0.095_{\pm 9.42e-03}$ ▲	$31.89_{\pm 3.20e+00}$ ▲
	gGA	$0.137_{\pm 1.46e-02}$ ▽	$0.106_{\pm 1.70e-02}$	$13.09_{\pm 2.11e+00}$
	scGA	$0.110_{\pm 2.10e-02}$ -	$0.110_{\pm 2.10e-02}$	$0.299_{\pm 4.59e-01}$
	ssGA	$0.118_{\pm 2.16e-02}$	$0.118_{\pm 2.17e-02}$ -	$0.103_{\pm 2.98e-01}$ ▽
256B	rdGA	$0.062_{\pm 5.50e-03}$	$0.051_{\pm 5.94e-03}$ ▲	$19.82_{\pm 2.26e+00}$ ▲
	gGA	$0.072_{\pm 6.41e-03}$ -	$0.060_{\pm 5.35e-03}$	$8.285_{\pm 1.20e+00}$
	scGA	$0.062_{\pm 6.58e-03}$ -	$0.062_{\pm 6.59e-03}$	$0.537_{\pm 6.08e-01}$
	ssGA	$0.067_{\pm 1.01e-02}$	$0.067_{\pm 1.01e-02}$ ▽	$0.024_{\pm 3.66e-02}$ ▽
1OAI	rdGA	$0.054_{\pm 2.75e-03}$	$0.039_{\pm 1.99e-03}$ ▲	$29.08_{\pm 2.10e+00}$ ▲
	gGA	$0.062_{\pm 5.41e-03}$ ▽	$0.043_{\pm 3.11e-03}$	$13.43_{\pm 2.20e+00}$
	scGA	$0.046_{\pm 8.20e-03}$ ▲	$0.046_{\pm 8.21e-03}$	$0.865_{\pm 1.02e+00}$
	ssGA	$0.049_{\pm 5.18e-03}$	$0.049_{\pm 5.20e-03}$ ▽	$0.265_{\pm 4.77e-01}$ ▽
1URR	rdGA	$0.115_{\pm 6.30e-03}$	$0.097_{\pm 4.96e-03}$ ▲	$22.96_{\pm 2.32e+00}$ ▲
	gGA	$0.124_{\pm 8.73e-03}$ ▽	$0.104_{\pm 5.89e-03}$	$8.830_{\pm 1.34e+00}$
	scGA	$0.103_{\pm 5.92e-03}$ ▲	$0.103_{\pm 5.93e-03}$	$0.255_{\pm 4.15e-01}$
	ssGA	$0.111_{\pm 8.45e-03}$	$0.111_{\pm 8.45e-03}$ ▽	$0.036_{\pm 1.63e-01}$ ▽

Table 10.1: Base study final value statistics for IFP samples



(a) Average fitness, 1B3A

(b) Average fitness, *1URR*

(c) Best fitness, *1B3A*

(d) Best fitness, *1URR*

(e) Average diversity, *1B3A*

(f) Average diversity, *1URR*

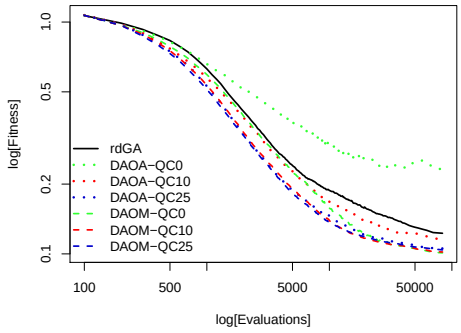
Figure 10.1: Convergence analysis of the base study on *1B3A* and *1URR*

10.3 Diversity as Objective with Quantile Constraint Study

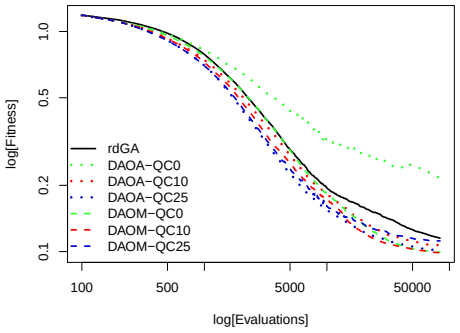
In the following the results of the rdGA and the DAO-QC NSGA-II with three different QC settings and two diversity objective variants are presented. Average and minimum Hamming distance diversity objective is denoted DAOA and DAOM respectively. Looking at Table 10.2 it is immediately clear that the DAOA-QC0 variant (i.e. DAO with average Hamming and without Quantile Constraint) is creating the most diversity of above 73%, significantly more than any of the others. However, this has significant impact on the quality of the solutions as well. The minimum Hamming counterpart, DAOM-QC0, is much more competitive and produces the best average fitness result sets for the shorter 1B3A and 1OAI proteins. Except for 256B, the three DAOM variants produce the best average, but this must be seen in the contrast of a very low diversity $\leq 10\%$. In general it is clear that the C_q setting has little effect on fitness achieved by the minimum Hamming diversity objective, DAOM variants. The explanation is found in the low diversity which is already less than optimal with DAOM-QC0. Higher C_q intensifies exploitation but exploration would be more helpful. This also explains why related works employing the minimal Hamming distance do not need the Quantile Constraint to be successful. With average Hamming diversity, i.e. DAOA, the Quantile Constraint is indeed required to "tame" the resulting diversity enhancing effects discussed on DAOA-QC0. With $C_q = 25$, the DAOA-QC25 achieves better results in terms of best fitness for all four protein samples with statistical significance, except on 256B.

		Average fitness	Best fitness	Average diversity
1B3A	rdGA	$0.122_{\pm 1.14e-02}$	$0.0948_{\pm 9.42e-03}$	$31.886_{\pm 3.20e+00}$
	DAOA-QC0	$0.226_{\pm 9.64e-02}$ ▽	$0.1060_{\pm 7.75e-03}$ ▽	$79.05_{\pm 3.30e+00}$ ▲
	DAOA-QC10	$0.116_{\pm 1.15e-02}$	$0.0895_{\pm 4.14e-03}$	$70.49_{\pm 4.63e+00}$
	DAOA-QC25	$0.106_{\pm 7.35e-03}$	$0.0862_{\pm 4.36e-03}$ ▲	$50.42_{\pm 6.08e+00}$
	DAOM-QC0	$0.100_{\pm 2.02e-02}$ -	$0.0929_{\pm 9.29e-03}$	$9.215_{\pm 1.17e+01}$
	DAOM-QC10	$0.102_{\pm 1.99e-02}$	$0.0997_{\pm 2.00e-02}$	$5.932_{\pm 1.30e+00}$ -
	DAOM-QC25	$0.104_{\pm 1.71e-02}$	$0.0995_{\pm 1.72e-02}$	$6.531_{\pm 1.22e+00}$
256B	rdGA	$0.0621_{\pm 5.50e-03}$	$0.0512_{\pm 5.94e-03}$	$19.82_{\pm 2.26e+00}$
	DAOA-QC0	$0.144_{\pm 1.11e-01}$ ▽	$0.0639_{\pm 4.06e-03}$ ▽	$73.05_{\pm 3.22e+00}$ ▲
	DAOA-QC10	$0.067_{\pm 3.91e-02}$	$0.0483_{\pm 3.26e-03}$	$58.15_{\pm 6.47e+00}$
	DAOA-QC25	$0.056_{\pm 4.20e-03}$ -	$0.0475_{\pm 3.82e-03}$ -	$36.49_{\pm 6.29e+00}$
	DAOM-QC0	$0.056_{\pm 6.76e-03}$	$0.0541_{\pm 6.51e-03}$	$5.490_{\pm 1.18e+00}$
	DAOM-QC10	$0.056_{\pm 7.73e-03}$	$0.0551_{\pm 7.55e-03}$	$4.428_{\pm 9.44e-01}$
	DAOM-QC25	$0.057_{\pm 5.64e-03}$	$0.0552_{\pm 5.42e-03}$	$4.331_{\pm 7.66e-01}$ -
1OAI	rdGA	$0.0543_{\pm 2.75e-03}$	$0.0387_{\pm 1.99e-03}$	$29.08_{\pm 2.10e+00}$
	DAOA-QC0	$0.175_{\pm 1.03e-01}$ ▽	$0.0481_{\pm 2.47e-03}$ ▽	$79.58_{\pm 2.37e+00}$ ▲
	DAOA-QC10	$0.062_{\pm 6.26e-03}$	$0.0398_{\pm 2.04e-03}$	$72.47_{\pm 3.08e+00}$
	DAOA-QC25	$0.053_{\pm 3.80e-03}$	$0.0368_{\pm 1.66e-03}$ ▲	$51.98_{\pm 5.24e+00}$
	DAOM-QC0	$0.041_{\pm 3.49e-03}$ -	$0.0381_{\pm 2.56e-03}$	$9.054_{\pm 1.93e+00}$
	DAOM-QC10	$0.041_{\pm 3.83e-03}$	$0.0399_{\pm 3.55e-03}$	$7.629_{\pm 1.92e+00}$
	DAOM-QC25	$0.044_{\pm 4.31e-03}$	$0.0418_{\pm 3.91e-03}$	$7.535_{\pm 7.91e-01}$ -
1URR	rdGA	$0.115_{\pm 6.30e-03}$	$0.0971_{\pm 4.96e-03}$	$22.96_{\pm 2.32e+00}$
	DAOA-QC0	$0.218_{\pm 2.49e-02}$ ▽	$0.1140_{\pm 5.20e-03}$ ▽	$73.79_{\pm 1.82e+00}$ ▲
	DAOA-QC10	$0.107_{\pm 5.31e-03}$	$0.0926_{\pm 2.99e-03}$	$59.64_{\pm 3.70e+00}$
	DAOA-QC25	$0.101_{\pm 4.60e-03}$	$0.0895_{\pm 3.90e-03}$ ▲	$33.98_{\pm 5.13e+00}$
	DAOM-QC0	$0.099_{\pm 4.41e-03}$	$0.0957_{\pm 4.00e-03}$	$5.280_{\pm 1.12e+00}$
	DAOM-QC10	$0.098_{\pm 5.77e-03}$ -	$0.0968_{\pm 5.61e-03}$	$3.948_{\pm 8.77e-01}$ ▽
	DAOM-QC25	$0.110_{\pm 2.80e-02}$	$0.1010_{\pm 6.45e-03}$	$6.118_{\pm 8.81e+00}$

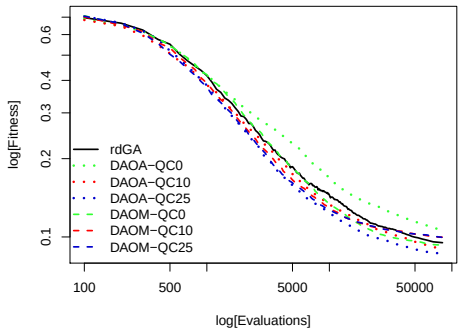
Table 10.2: DAO-QC study final value statistics for IFP samples



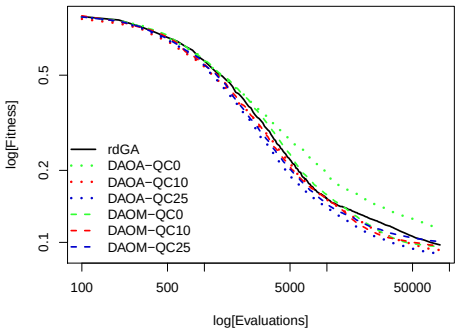
(a) Average fitness, 1B3A



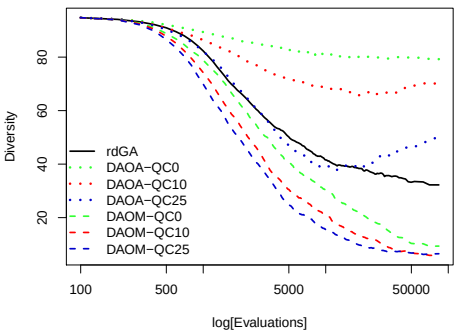
(b) Average fitness, *1URR*



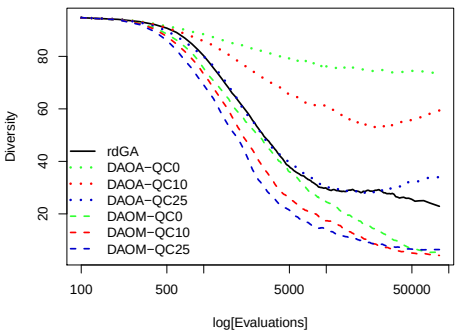
(c) Best fitness, *1B3A*



(d) Best fitness, *1URR*



(e) Average diversity, *1B3A*



(f) Average diversity, *1URR*

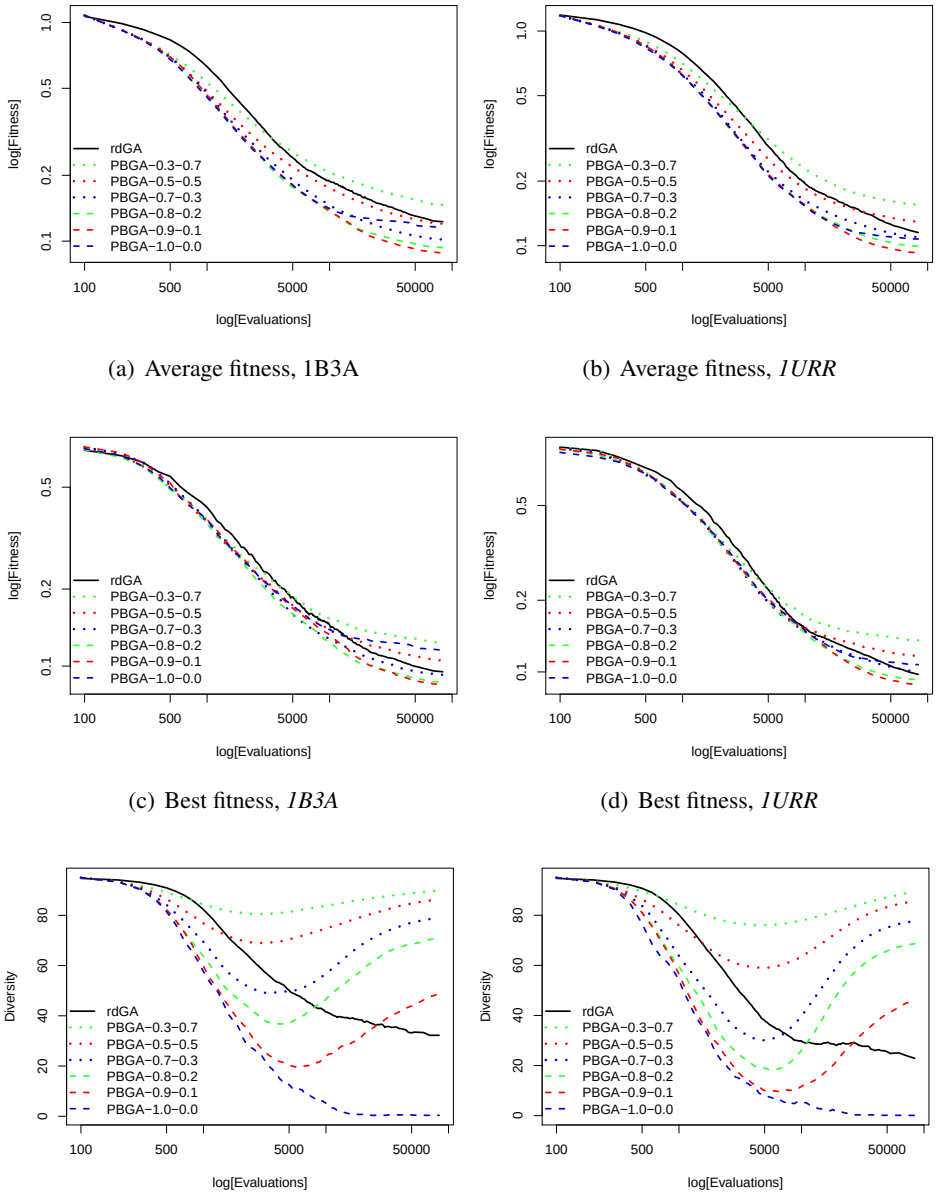
Figure 10.2: Convergence analysis of the DAO-QC study on *1B3A* and *1URR*

10.4 Preference Based Genetic Algorithm Study

In the following the results of the rdGA and the PBGA with six weight settings corresponding to different amounts of diversity preference are studied. Like the DAOA-QC algorithm tested in Section 10.3 the PBGA is able to achieve any level of diversity. From Figure 10.3(a-f) and Table 10.3 it is obvious that the golden setting seems to be $W_{(fit,div)} = (0.9, 0.1)$. The PBGA achieves the best average and best fitness with statistical significance for all four protein samples. At the same time the diversity reaches values around 50% with an interesting progress in Figure 10.3(e-f). Around the bend in the fitness curve at $5-8000$ evaluations, which was also present in the other experiments, the diversity curve starts increasing again. It suggests that after transitioning into a converged state where fitness progress is generated by random sampling of the mutation operator, the algorithm starts to diversify again. This is achieved by the algorithm population and problem characteristics without any form of active adaptive mechanism.

		Average fitness	Best fitness	Average diversity
1B3A	rdGA	$0.122_{\pm 1.14e-02}$	$0.0948_{\pm 9.42e-03}$	$31.886_{\pm 3.20e+00}$
	PBGA 0.3 0.7	$0.146_{\pm 3.08e-03}$ ▽	$0.124_{\pm 3.50e-03}$ ▽	$89.936_{\pm 1.83e-01}$ ▲
	PBGA 0.5 0.5	$0.120_{\pm 3.06e-03}$	$0.105_{\pm 3.50e-03}$	$86.234_{\pm 3.34e-01}$
	PBGA 0.7 0.3	$0.102_{\pm 2.75e-03}$	$0.0924_{\pm 2.83e-03}$	$78.870_{\pm 7.28e-01}$
	PBGA 0.8 0.2	$0.0932_{\pm 2.39e-03}$	$0.0867_{\pm 2.50e-03}$	$70.961_{\pm 1.08e+00}$
	PBGA 0.9 0.1	$0.0883_{\pm 4.10e-03}$ ▲	$0.0846_{\pm 3.86e-03}$ ▲	$49.026_{\pm 2.80e+00}$
	PBGA 1.0 0.0	$0.115_{\pm 2.05e-02}$	$0.115_{\pm 2.02e-02}$	$0.342_{\pm 6.47e-01}$ ▽
256B	rdGA	$0.0621_{\pm 5.50e-03}$	$0.0512_{\pm 5.94e-03}$	$19.815_{\pm 2.26e+00}$
	PBGA 0.3 0.7	$0.110_{\pm 2.09e-03}$ ▽	$0.0883_{\pm 3.44e-03}$ ▽	$90.730_{\pm 1.79e-01}$ ▲
	PBGA 0.5 0.5	$0.0819_{\pm 1.71e-03}$	$0.0688_{\pm 2.12e-03}$	$87.296_{\pm 2.96e-01}$
	PBGA 0.7 0.3	$0.0645_{\pm 1.92e-03}$	$0.0561_{\pm 1.91e-03}$	$79.947_{\pm 5.30e-01}$
	PBGA 0.8 0.2	$0.0561_{\pm 2.63e-03}$	$0.0503_{\pm 2.52e-03}$	$72.175_{\pm 8.64e-01}$
	PBGA 0.9 0.1	$0.0501_{\pm 3.86e-03}$ ▲	$0.0466_{\pm 3.87e-03}$ ▲	$53.132_{\pm 2.04e+00}$
	PBGA 1.0 0.0	$0.0653_{\pm 6.82e-03}$	$0.0653_{\pm 6.82e-03}$	$0.291_{\pm 3.79e-01}$ ▽
1OAI	rdGA	$0.0543_{\pm 2.75e-03}$	$0.0387_{\pm 1.99e-03}$	$29.084_{\pm 2.10e+00}$
	PBGA 0.3 0.7	$0.0947_{\pm 1.23e-03}$ ▽	$0.0755_{\pm 3.55e-03}$ ▽	$91.689_{\pm 1.11e-01}$ ▲
	PBGA 0.5 0.5	$0.0704_{\pm 8.30e-04}$	$0.058_{\pm 2.14e-03}$	$88.159_{\pm 1.86e-01}$
	PBGA 0.7 0.3	$0.0547_{\pm 1.15e-03}$	$0.0468_{\pm 1.94e-03}$	$81.727_{\pm 4.82e-01}$
	PBGA 0.8 0.2	$0.0473_{\pm 1.32e-03}$	$0.0411_{\pm 1.63e-03}$	$74.866_{\pm 8.19e-01}$
	PBGA 0.9 0.1	$0.0395_{\pm 1.43e-03}$ ▲	$0.0355_{\pm 1.59e-03}$ ▲	$58.162_{\pm 1.74e+00}$
	PBGA 1.0 0.0	$0.0472_{\pm 7.15e-03}$	$0.0472_{\pm 7.15e-03}$	$0.432_{\pm 4.85e-01}$ ▽
1URR	rdGA	$0.115_{\pm 6.30e-03}$	$0.0971_{\pm 4.96e-03}$	$22.962_{\pm 2.32e+00}$
	PBGA 0.3 0.7	$0.155_{\pm 1.85e-03}$ ▽	$0.136_{\pm 3.46e-03}$ ▽	$89.186_{\pm 1.77e-01}$ ▲
	PBGA 0.5 0.5	$0.129_{\pm 2.05e-03}$	$0.117_{\pm 3.17e-03}$	$85.503_{\pm 2.42e-01}$
	PBGA 0.7 0.3	$0.109_{\pm 2.21e-03}$	$0.100_{\pm 2.51e-03}$	$77.668_{\pm 5.33e-01}$
	PBGA 0.8 0.2	$0.0991_{\pm 2.11e-03}$	$0.0929_{\pm 2.23e-03}$	$68.759_{\pm 8.30e-01}$
	PBGA 0.9 0.1	$0.0924_{\pm 5.14e-03}$ ▲	$0.0885_{\pm 4.89e-03}$ ▲	$46.444_{\pm 3.18e+00}$
	PBGA 1.0 0.0	$0.107_{\pm 9.52e-03}$	$0.107_{\pm 9.52e-03}$	$0.116_{\pm 2.04e-01}$ ▽

Table 10.3: PBGA study final value statistics for IFP samples



(a) Average fitness, 1B3A

(b) Average fitness, *1URR*

(c) Best fitness, *1B3A*

(d) Best fitness, *1URR*

(e) Average diversity, *1B3A*

(f) Average diversity, *1URR*

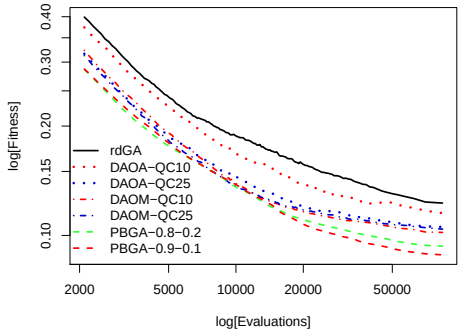
Figure 10.3: Convergence analysis of the PBGA study on *1B3A* and *1URR*

10.5 Summary of Algorithm Experiments

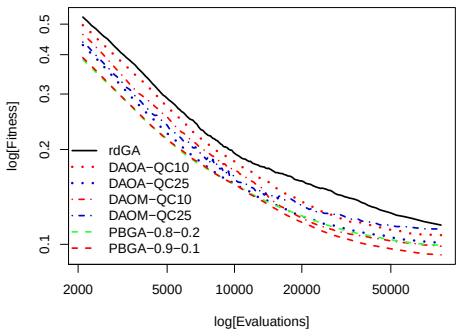
After establishing the superior performance of both developed algorithms DAO-QC and PBGA with respect to the standard algorithms, they are lined up head-to-head for a direct comparison. Usage of the best found settings in the respective algorithm variants, deducted in the previous experiments, should provide a fair ground for comparison. The overall picture shown in Figure 10.4 shows quite clearly that the PBGA achieves a better performance on all protein samples with the weight of $W_{(fit,div)} = (0.9, 0.1)$. The results are supported by numbers in Table 10.4, where the PBGA is seen to produce the best average fitness with statistical significance for all four protein samples. The same is for the best fitness measure, where significance is only given for the 1OAI sample. The DAO-QC25 approach is a close second place in terms of best fitness, and can be regarded as statistically equivalent to the $W_{(fit,div)} = (0.9, 0.1)$ PBGA except for 1OAI. These results must also be seen in the light of a significantly higher diversity, except for 1B3A.

		Average fitness	Best fitness	Average diversity
1B3A	rdGA	$0.122_{\pm 1.14e-02}$ ▽	$0.0948_{\pm 9.42e-03}$	$31.886_{\pm 3.20e+00}$
	DAOA-QC10	$0.116_{\pm 1.15e-02}$	$0.0895_{\pm 4.14e-03}$	$70.49_{\pm 4.63e+00}$
	DAOA-QC25	$0.106_{\pm 7.35e-03}$	$0.0862_{\pm 4.36e-03}$	$50.42_{\pm 6.08e+00}$
	DAOM-QC10	$0.102_{\pm 1.99e-02}$	$0.0997_{\pm 2.00e-02}$ -	$5.932_{\pm 1.30e+00}$ -
	DAOM-QC25	$0.104_{\pm 1.71e-02}$	$0.0995_{\pm 1.72e-02}$	$6.531_{\pm 1.22e+00}$
	PBGA 0.8 0.2	$0.0932_{\pm 2.39e-03}$	$0.0867_{\pm 2.50e-03}$	$70.961_{\pm 1.08e+00}$ -
	PBGA 0.9 0.1	$0.0883_{\pm 4.10e-03}$ ▲	$0.0846_{\pm 3.86e-03}$ -	$49.026_{\pm 2.80e+00}$
256B	rdGA	$0.0621_{\pm 5.50e-03}$	$0.0512_{\pm 5.94e-03}$	$19.815_{\pm 2.26e+00}$
	DAOA-QC10	$0.067_{\pm 3.91e-02}$ -	$0.0483_{\pm 3.26e-03}$	$58.15_{\pm 6.47e+00}$
	DAOA-QC25	$0.056_{\pm 4.20e-03}$	$0.0475_{\pm 3.82e-03}$	$36.49_{\pm 6.29e+00}$
	DAOM-QC10	$0.056_{\pm 7.73e-03}$	$0.0551_{\pm 7.55e-03}$	$4.428_{\pm 9.44e-01}$
	DAOM-QC25	$0.057_{\pm 5.64e-03}$	$0.0552_{\pm 5.42e-03}$ -	$4.331_{\pm 7.66e-01}$ -
	PBGA 0.8 0.2	$0.0561_{\pm 2.63e-03}$	$0.0503_{\pm 2.52e-03}$	$72.175_{\pm 8.64e-01}$ ▲
	PBGA 0.9 0.1	$0.0501_{\pm 3.86e-03}$ ▲	$0.0466_{\pm 3.87e-03}$ -	$53.132_{\pm 2.04e+00}$
1OAI	rdGA	$0.0543_{\pm 2.75e-03}$	$0.0387_{\pm 1.99e-03}$	$29.084_{\pm 2.10e+00}$
	DAOA-QC10	$0.062_{\pm 6.26e-03}$ ▽	$0.0398_{\pm 2.04e-03}$	$72.47_{\pm 3.08e+00}$
	DAOA-QC25	$0.053_{\pm 3.80e-03}$	$0.0368_{\pm 1.66e-03}$	$51.98_{\pm 5.24e+00}$
	DAOM-QC10	$0.041_{\pm 3.83e-03}$	$0.0399_{\pm 3.55e-03}$	$7.629_{\pm 1.92e+00}$
	DAOM-QC25	$0.044_{\pm 4.31e-03}$	$0.0418_{\pm 3.91e-03}$ -	$7.535_{\pm 7.91e-01}$ -
	PBGA 0.8 0.2	$0.0473_{\pm 1.32e-03}$	$0.0411_{\pm 1.63e-03}$	$74.866_{\pm 8.19e-01}$ ▲
	PBGA 0.9 0.1	$0.0395_{\pm 1.43e-03}$ ▲	$0.0355_{\pm 1.59e-03}$ ▲	$58.162_{\pm 1.74e+00}$
1URR	rdGA	$0.115_{\pm 6.30e-03}$ ▽	$0.0971_{\pm 4.96e-03}$	$22.962_{\pm 2.32e+00}$
	DAOA-QC10	$0.107_{\pm 5.31e-03}$	$0.0926_{\pm 2.99e-03}$	$59.64_{\pm 3.70e+00}$
	DAOA-QC25	$0.101_{\pm 4.60e-03}$	$0.0895_{\pm 3.90e-03}$	$33.98_{\pm 5.13e+00}$
	DAOM-QC10	$0.098_{\pm 5.77e-03}$	$0.0968_{\pm 5.61e-03}$	$3.948_{\pm 8.77e-01}$ ▽
	DAOM-QC25	$0.110_{\pm 2.80e-02}$	$0.1010_{\pm 6.45e-03}$ ▽	$6.118_{\pm 8.81e+00}$
	PBGA 0.8 0.2	$0.0991_{\pm 2.11e-03}$	$0.0929_{\pm 2.23e-03}$	$68.759_{\pm 8.30e-01}$ ▲
	PBGA 0.9 0.1	$0.0924_{\pm 5.14e-03}$ ▲	$0.0885_{\pm 4.89e-03}$ -	$46.444_{\pm 3.18e+00}$

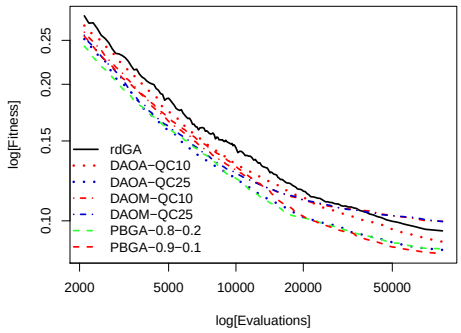
Table 10.4: PBGA vs. DAO-QC study final value statistics for IFP samples



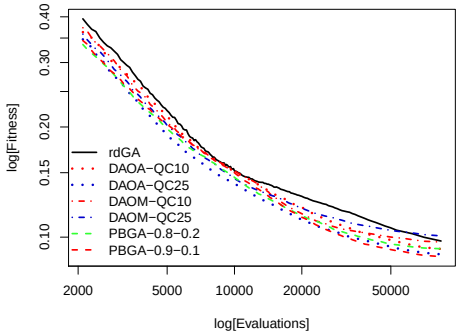
(a) Average fitness, 1B3A



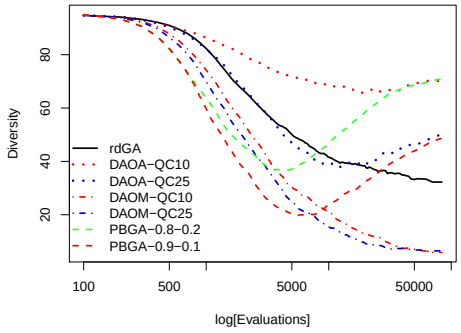
(b) Average fitness, *1URR*



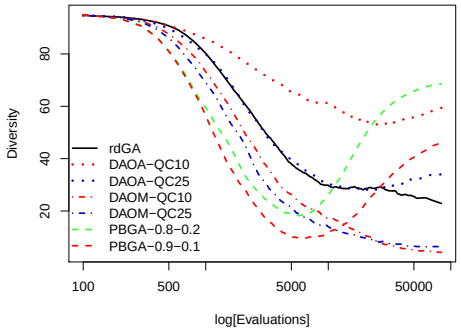
(c) Best fitness, *1B3A*



(d) Best fitness, *1URR*



(e) Average diversity, *1B3A*

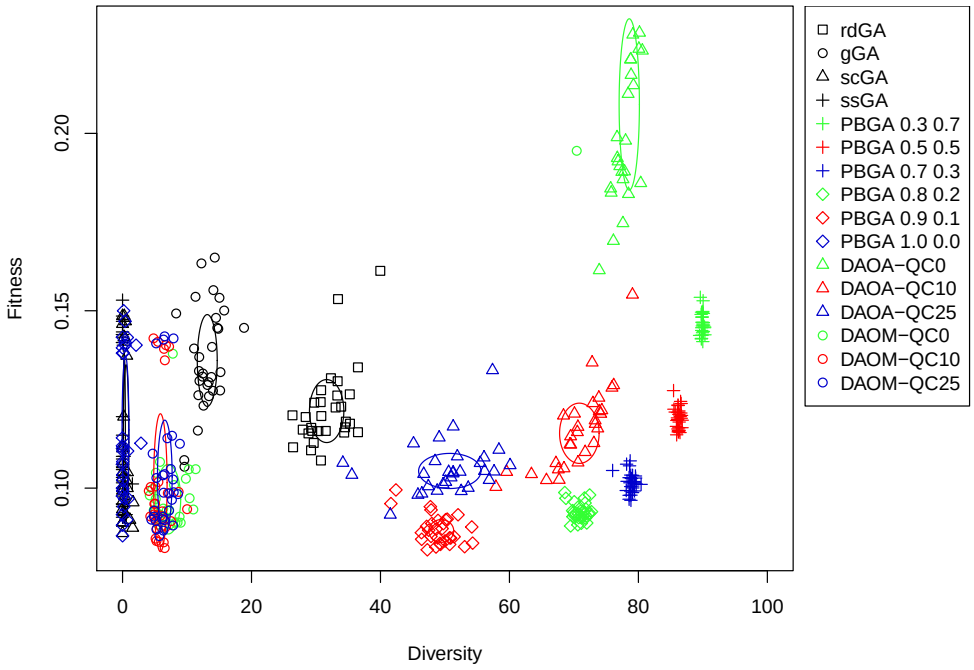


(f) Average diversity, *1URR*

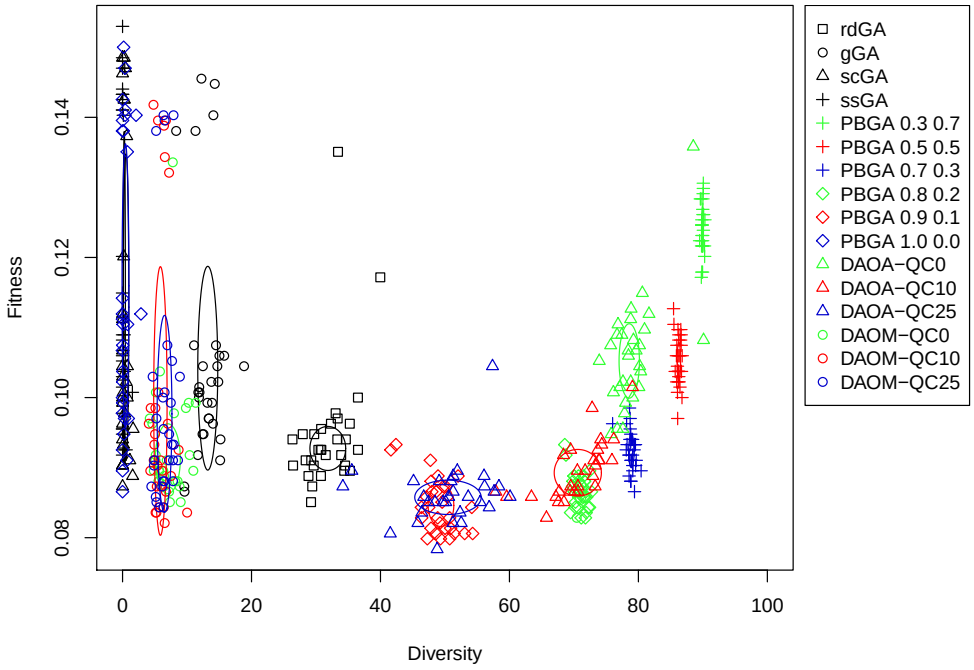
Figure 10.4: Convergence analysis of the PBGA vs. DAO study on *1B3A* and *1URR*

10.5.1 Diversity-Fitness Study

To provide a global view on the exploration vs. exploitation trade-off, the complete set of final values for all algorithms are shown in a single plot. This allows to situate the overall performance of the algorithms according to each other, but also to visualise the variation in results from run to run. Figures 10.5 and 10.6 show these plots for protein samples 1B3A and 1URR. The better performance of the PBGA 0.9 0.1 in both best and average fitness is confirmed by the cluster of results situated lower and more to the right than the DAOA-QC25. In addition, the plots show that the reliability of the PBGA is higher with less risk of a optimisation run of failing with a poor result. This fact is deducted from the compact clusters representing different algorithm settings. The DAO-QC on the other hand produces results with greater variation from run to run both in terms of fitness and diversity. Nevertheless the DAOA-QC25 and the PBGA 0.9 0.1 are statistically equivalent for the 1B3A sample on best fitness, and the DAO approach finds the best solutions among the $2x30$. The DAO approach is also interesting to study as a reference because other methods in literature employ the DAO approach with Minimum Hamming without Quantile Constraint, which in the case of the IFP is not the best choice.

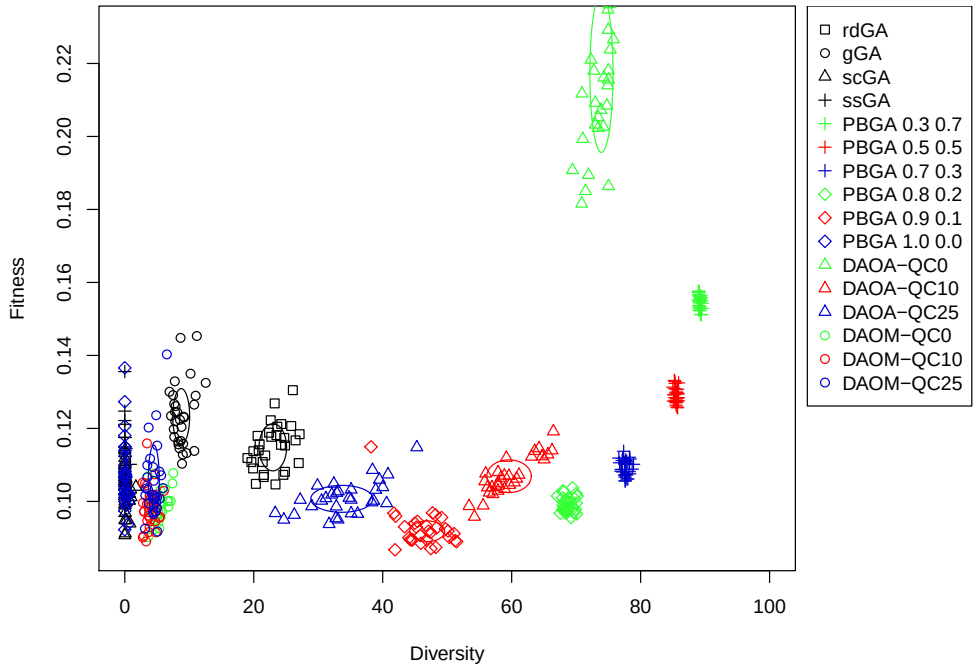


(a) Average fitness

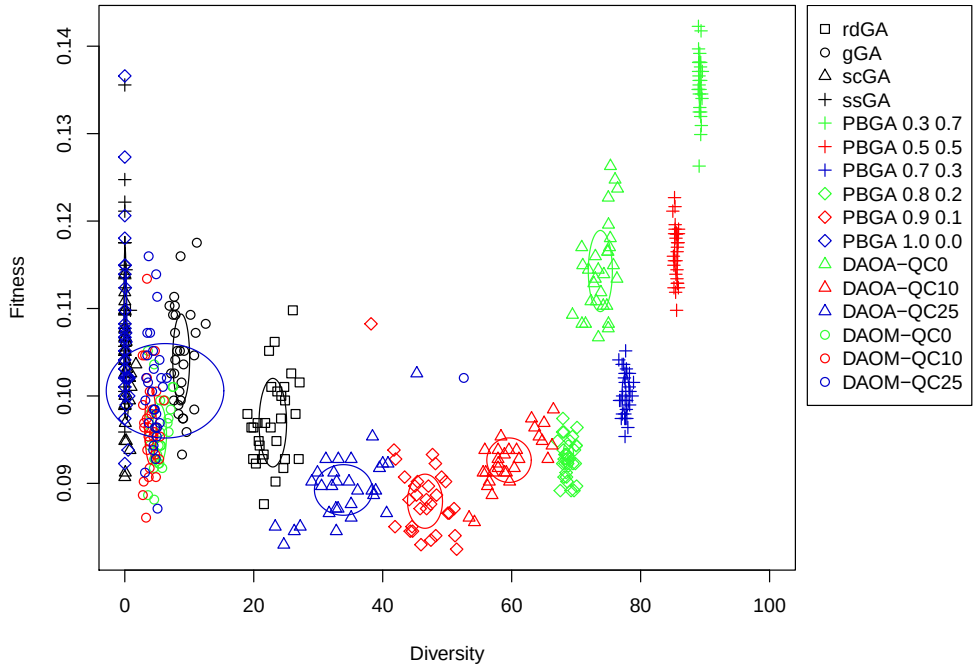


(b) Best fitness

Figure 10.5: Diversity vs. average fitness of all algorithms on 1B3A. Each point represents an individual run.



(a) Average fitness



(b) Best fitness

Figure 10.6: Diversity vs. average fitness of all algorithms on 1URR. Each point represents an individual run.

10.6 Conclusions

As initial investigation, a baseline study was performed to get a first view of the general performance of existing algorithms on the IFP problem. The remove doubles GA (rdGA) was established as the better performing of the base study algorithms, generational GA (gGA), steady state GA (ssGA) and synchronous cellular GA (scGA). Further, the removal of doubles feature induces a good diversity level, bringing it closer to the diversity preserving algorithms presented in this work. Both the developed algorithms, DAO-QC and PBGA, were able to outperform the rdGA in fitness and diversity at the same time. When compared, each using the best settings found, the PBGA shows a slight advantage over the DAO-QC approach. Though for many IFP instances their performance in terms of best fitness is equivalent, with statistical significance, the PBGA has slightly better results on average, and at the same time the diversity is at a higher level. By aggregating all results in the Diversity-Fitness plots it is also clear that the PBGA produce results in a more narrow region of the plot for a given set of settings. The DAO-QC on the contrary, shows a larger variation of the results from tun to run, but this variation also caused it to find the best solution to the 1B3A protein.

11

Experiments on NKL Benchmark

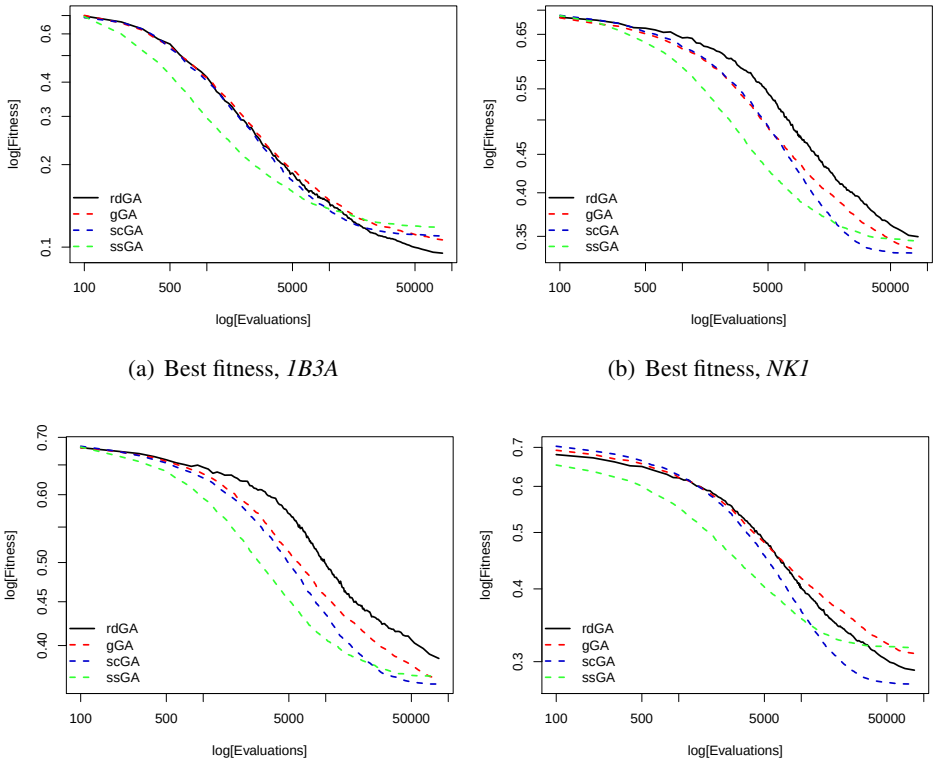
11.1 Introduction

In this chapter the various algorithms are tested on the developed benchmark instances following the same plan as for the real proteins in Chapter 10 and with the setup described in Chapter 9. The impact that changing of algorithms and their settings has on the resulting performance will be presented. With the conclusion of the landscape analysis of the NKL benchmark model presented in Chapter 7 in mind, the following experiments aim at further demonstrating analogies between the Inverted Folding Problem model and the benchmark.

11.2 Baseline Study

Landscape analysis shows that the sub-sampling needs to be matched the length of the sample and the NK instance. With respect to the adaptive walk tests, a sub-sampling value P of 4 and 5 produced the closest matched walk length for the two samples respectively with NK Model 1. For Model 2 the sub-sampling of $P = 4$ was not enough to reach the same walk lengths. Where the shorter Model 1 needed less sub-sampling Model 2 needed as low as $P = 5$ to match the auto-correlation walk lengths. For the real protein instances the rdGA was performing the best, but on the NK model it is the worst as seen in Figure 11.1.

Figure 11.1 also shows that a low sub-sampling of $P = 4$ has the desired effect of mimicking the IFP better, and the rdGA now becomes the second best algorithm, overtaking gGa and ssGA. Therefore, the rest of the convergence plots of this chapter will emphasise the NK Model instances with $P = 4$. Please refer to the Appendix Section B.2 for the remaining plots and statistics.



(a) Best fitness, *1B3A*

(b) Best fitness, *NK1*

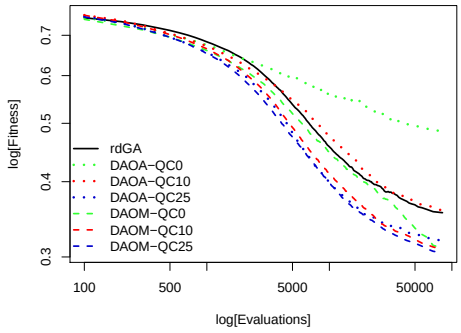
(c) Best fitness, *NK2*

(d) Best fitness, *NK1*, $P = 4$

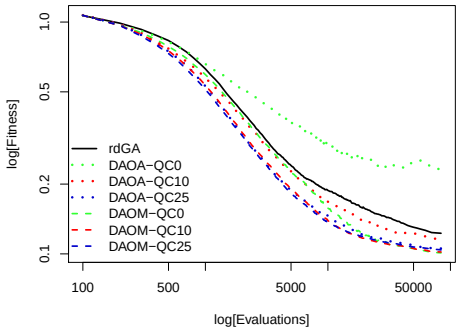
Figure 11.1: Convergence analysis of the base study on *Model* and *1B3A*

11.3 Diversity as Objective with Quantile Constraint Study

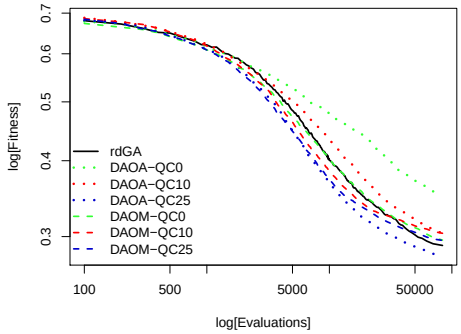
The results of the experiments employing the DAO-QC NSGA-II algorithm on the Benchmark Models are summarised in the following and compared to the standard GA for baseline comparison. With 1B3A as an example, Figure 11.2 shows a direct comparison of the IFP Model and the analogous NK Model benchmark. The NK Model manages to produce the same order of the results when inspecting average Hamming runs. The min Hamming approach is hardly affected by the quantile Constraint setting, and establishing an order is difficult. But generally comparing DAOM and DAOA-QC25 with respect to best and average fitness, the order is correctly emulated. Concerning diversity, the two plots look very similar and all data series are arranged in the same order. In addition, features such as the slight increase in diversity visible for the DAOA runs, are replicated and happen approximately at the same point in time. One slight difference to point out is the lower diversity induced in the NK runs by the same C_q setting. Likewise for the fitness plot the transition to a flatter slope happens gradually above 10000, and the results on the IFP display a larger spread, may be a desired effect when the goal is to find and design the most efficient algorithm for the problem. Overall many features observed on the IFP instance experiments are well reproduced and one could easily be lead to think that the plots originate from two different instances of the same problem, rather than from a benchmark and a real-world problem.



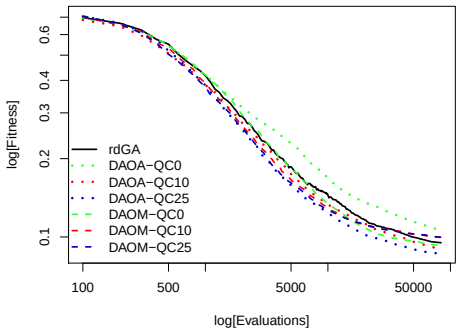
(a) Average fitness, *NK1*, $P = 4$



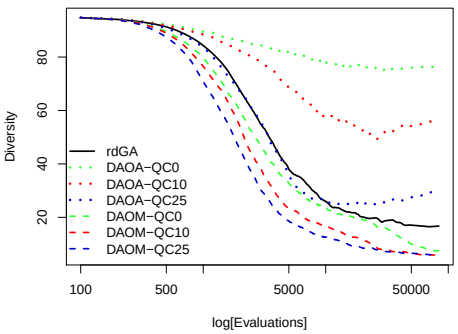
(b) Average fitness, *1B3A*



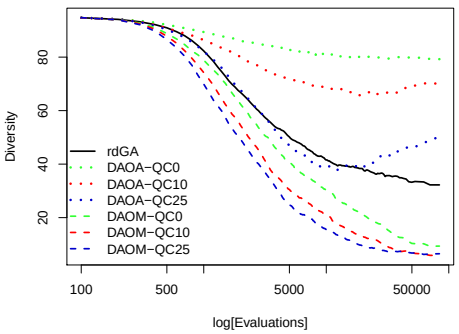
(c) Best fitness, *NK1*, $P = 4$



(d) Best fitness, *1B3A*



(e) Average diversity, *NK1*, $P = 4$

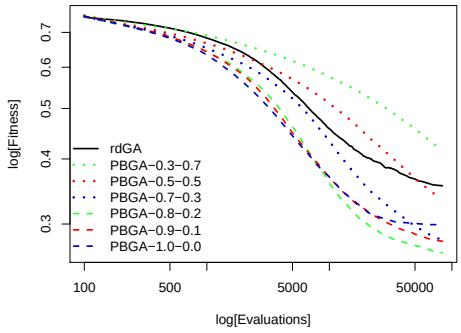


(f) Average diversity, *1B3A*

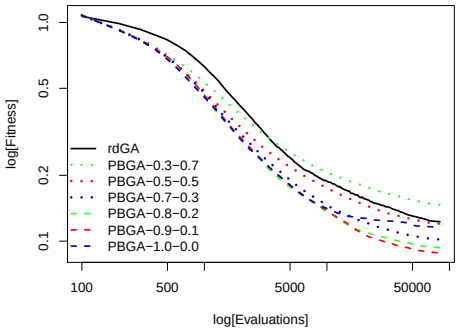
Figure 11.2: Convergence analysis of the DAO-QC study on *NK1* and *1B3A*

11.4 Preference Based Genetic Algorithm Study

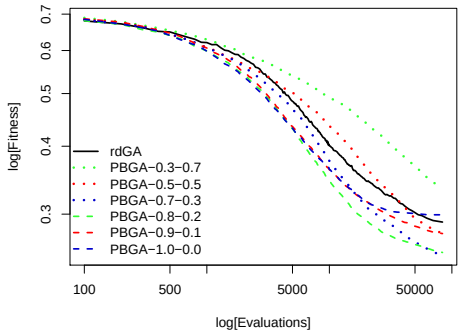
The results of the experiments employing the PBGA algorithm on the Benchmark Models are summarised in the following and compared to the standard GA for baseline comparison. Like for the DAO approach discussed in the previous section, very similar characteristics are achieved both in terms of fitness and diversity when compared to the 1B3A runs, see Figure 11.3. The average fitness plots show a similar ordering with the exception of the $W_{(fit,div)} = (0.9, 0.1)$ and $W_{(fit,div)} = (0.8, 0.2)$ swapping places. The golden ratio seems to be between $(0.7, 0.3)$ and $(0.8, 0.2)$ and the former is performing best on the best fitness plot. The explanation has its roots in an observation done in the previous section with the DAO-QC plots: more emphasis on exploration is required to achieve and maintain the same amount of diversity. The effect is also very visible in Figure 11.3(e,f), and a very interesting observation to point out is: the best performance in fitness is always achieved by the approach that maintains a diversity of just below 40%, also when looking back at Figure 11.2. Hence, in this regard, the NK Model can be said to capture another important characteristic of the IFP, that of simulating what level of diversity produces the optimal performance.



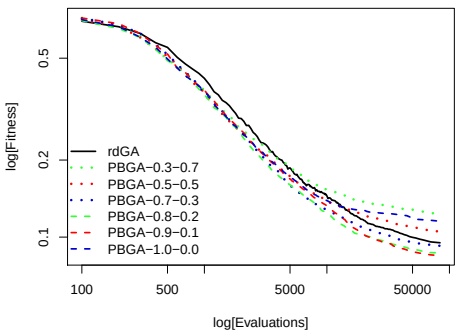
(a) Average fitness, *NK1*, $P = 4$



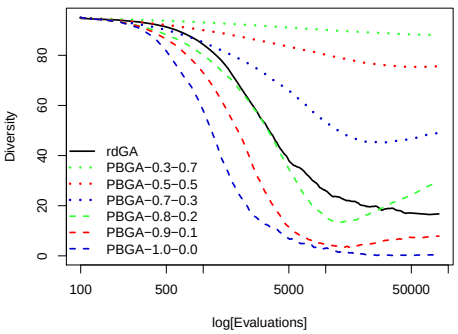
(b) Average fitness, *1B3A*



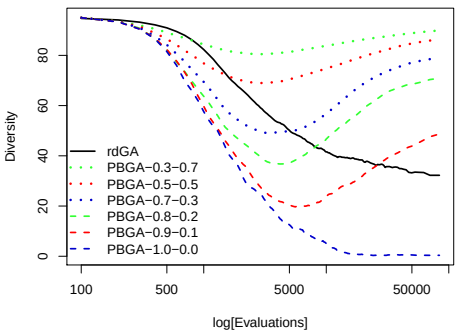
(c) Best fitness, *NK1*, $P = 4$



(d) Best fitness, *1B3A*



(e) Average diversity, *NK1*, $P = 4$



(f) Average diversity, *1B3A*

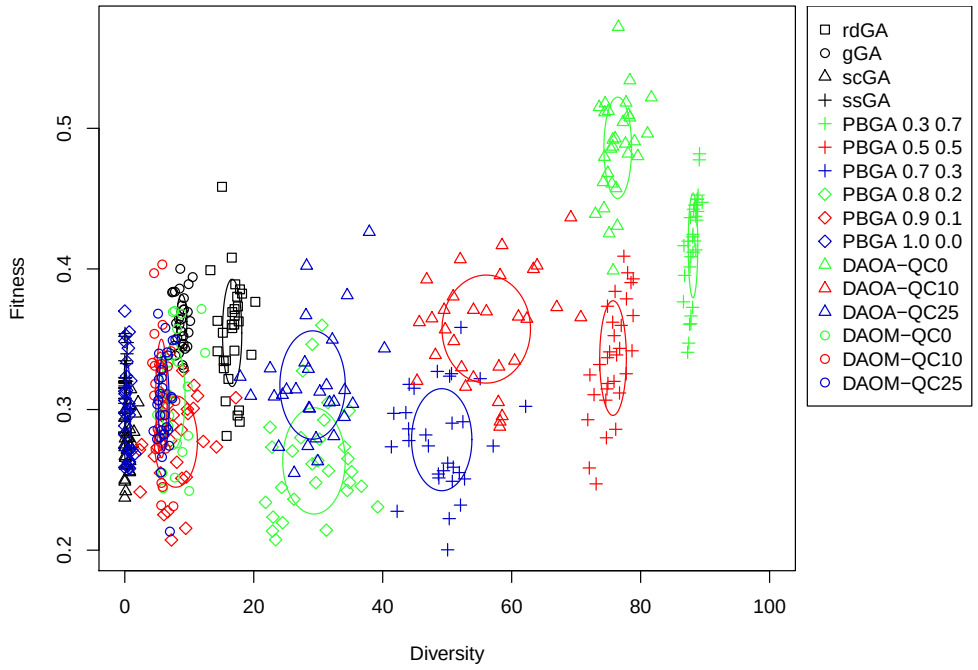
Figure 11.3: Convergence analysis of the PBGA study on *NK1* and *1B3A*

11.5 Summary of Algorithm Experiments Study

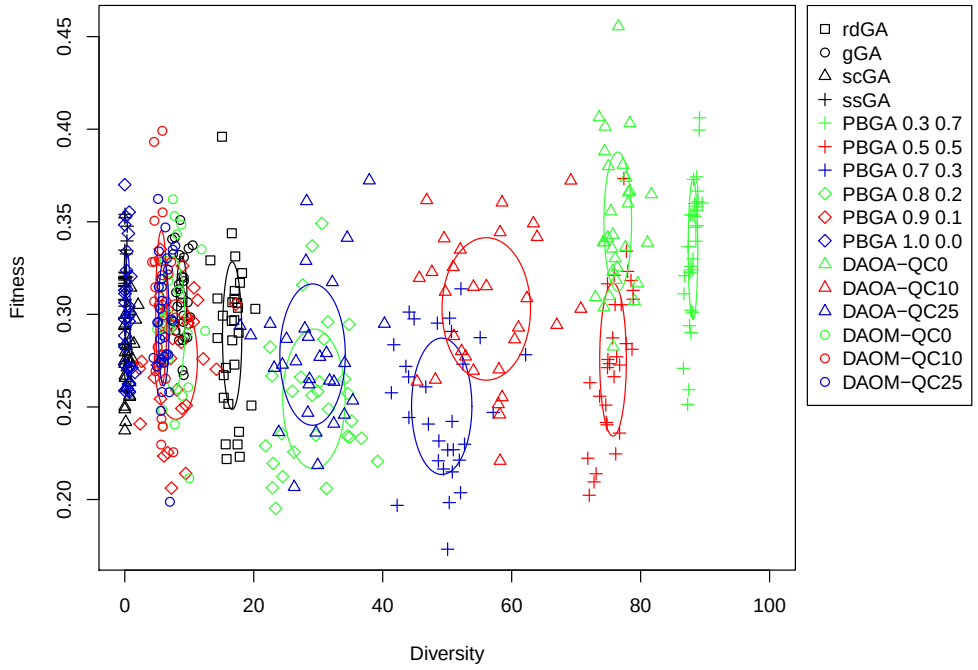
In the summary study of the IFP problem in Section 10.5, the developed algorithms DAO-QC and PBGA are sought compared head-to-head on benchmark instances using the same settings. It is clear, referring to Sections 11.3 and 11.4, that on NK Model benchmarks without sub-sampling the DAO-QC is performing significantly better than its rdGA counterpart, but noticeably worse than the PBGA. Therefore this section will not go further into the details of comparing the algorithms DAO-QC and PBGA, but refers instead to the Section B.5 for full details.

11.5.1 Diversity-Fitness Study

To provide a global view on the exploration vs. exploitation performance of the different algorithms tested on the NK Model the a diversity-fitness study of Section 10.5.1 is repeated hereafter. Figure 11.4 shows the example of the NK1 Model with sub-sampling $P = 4$ to mimic IFP sample 1B3A the best possible. The equivalent diagram is shown in Figure 10.5 where the clusters corresponding to the different algorithms and settings are evident. There is a greater spread though in the points than for the IFP Model counterpart.



(a) Average fitness



(b) Best fitness

Figure 11.4: Diversity vs. average fitness of all algorithms on Model 1 with subsampling $P = 4$. Each point represents an individual run.

11.6 Conclusions

The same batch of studies that were conducted on the IFP instances in Chapter 10 were repeated on the NK Model to test different configuration settings. It was demonstrated through selected plots and statistics that very similar characteristics can indeed be achieved with the right configuration of the NK Model. The higher ruggedness stemming from the higher number of epistatic links of the NK2 Model seems to be a little bit too strong compared to the less rugged NK1 Model. Highly similar characteristics were demonstrated between the IFP sample 1B3A and the NK1 Model with $N = 67$ and $P = 4$. In the base-study, the same ordering of algorithms rdGA, gGA. scGA and ssGA was achieved in terms of average fitness and diversity. For both developed algorithms NSGA-II DAO-QC and PBGA the settings that performed the best on the IFP were also close to the best setting of the NK Model, which is desirable feature for algorithm designers. In all experiments on DAO-QC and PBGA the NK Model produced slightly less population diversity than the real IFP given the same diversity preserving setting. At the same time it was noticed that the best performance was always achieved by the approach that maintained a diversity of just below 40% roughly. Hence, in this regard, the NK Model can be said to capture another important characteristic of the IFP, that of simulating what level of diversity produces the optimal performance.

12

Validation Through Structure Prediction

12.1 Structure Validation

In this second experimental step, the protein sequences generated by the best performing algorithm are validated. To this end the I-TASSER[75] prediction tool is used to generate their secondary and tertiary structures that will be compared to the structure of the targeted protein. For each sample, the five best generated sequences of the final population in each of the 30 individual runs are selected. This means a total of 300 I-TASSER runs for the two protein samples, each run taking around two days, which amounts to almost two years of CPU-time. It is to be noted that the I-TASSER prediction itself is subject to erroneous results, hence a 100% certainty can never be achieved unless the proteins are synthesised in a wet-lab. In the following, the sequences and their I-TASSER predictions are analysed in terms of primary and secondary structure in section 12.1.1, and then tertiary structure in section 12.1.2. The full workflow executed to obtain the values of the study is shown in Figure 12.1.

12.1.1 Primary and Secondary Structure Validation Results

The goal in this section is to analyse how well the secondary structure of the reference protein is reproduced in the predicted model and is based on the *Secondary Match* and *Identity* output in the workflow (see Figure 12.1). Table 12.1 shows a summary of the two proteins tested. Clearly, the generated sequences share very little resemblance with the original input sequence seen from a sequence identity of about 20% and 15% respectively with a very low deviation. Achieving low sequence identity by itself is not a challenging task unless a good structure match is obtained at the same time. The table shows this as the average percentage, μ, of positions in the secondary annotation of the I-TASSER predicted model that correctly matches those of the input annotation. Average percentage μ and standard deviation of the average percentage σ is given for each of the three structure types H, E and L. As it can be seen the

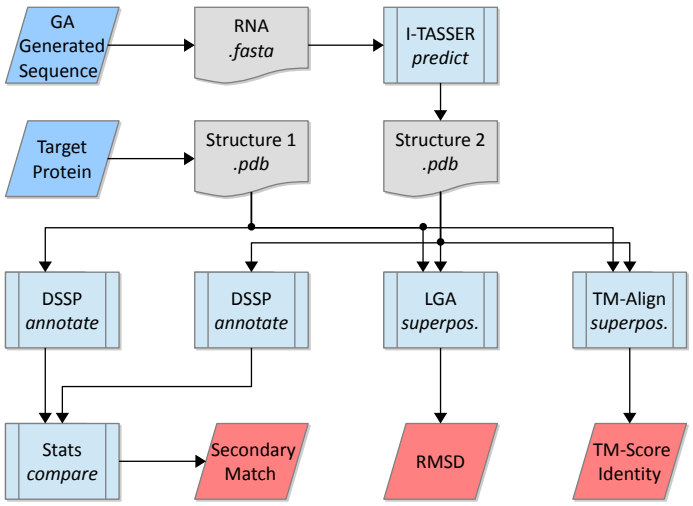


Figure 12.1: The workflow involved in generating the different validation values.

helices are correctly predicted on more than 90% of the positions in both proteins. For the slightly bigger *1URR* sample which contrary to *1OAI* contains many extended sheets the sheet match percentage is lower - slightly below 50%.

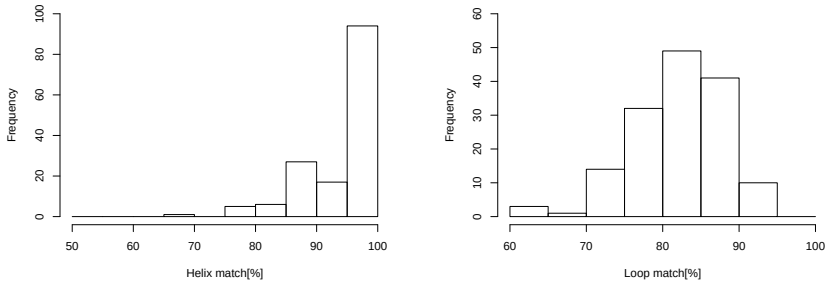
Figure 12.2 and Figure 12.3 illustrate the same data as histograms. Figure 12.2 (a) and Figure 12.3 (a) clearly demonstrate that *helix* structures are very well matched in all 300 structure predictions. Almost all of the tested generated individuals have a match-percentage of over 80% and the majority is above 90% for both samples.

For *loop* segments presented in Figure 12.2 (b) and Figure 12.3 (c), the majority is still above 80% but with a high spread. The statistics for *sheet* segments show that there is a limit to the performance of an approach optimising only an approximate secondary structure prediction. Considering that the *1URR* sample consists of 6 *sheet* segments across the whole of its length, then 42% can be considered as a rather good result. The lower success-rate of predicting sheets is due to the fact that a sheet can only be observed in the secondary structure if the I-TASSER predicted structure actually did fold close enough to the reference tertiary structure, to allow the extended sheet to form. A *Helix* is a much more local structure mostly independent of the global fold, hence easier to achieve in this analysis.

Table 12.1: Summary of secondary structure prediction match

Protein	$\mu_{Identity}$	$\sigma_{Identity}$	μ_{Helix}	σ_{Helix}	μ_{Sheet}	σ_{Sheet}	μ_{Loop}	σ_{Loop}
1OAI	20.67	4.100	93.348	6.343	0	0	82.814	7.368
1URR	15.23	3.225	93.787	6.563	42.108	8.898	85.523	8.239

Figure 12.4 and Table 12.3 show the alignment of three of the best aligned indi-



(a) *Helix* match histogram (b) *Loop* match histogram

Figure 12.2: Match histograms of *1OAI* - percentage of positions in the secondary annotation of the I-TASSER predicted model that correctly matches those of the input annotation.

vidually generated sequences. This is to show specific examples of the results which have been averaged in Table 12.1 and the tendency remains the same: *helices* are very well defined with above 95% positions matched, *loops* slightly less with ±90% and ±80% for samples *1OAI* and *1URR* respectively. The *1OAI* sample is clearly an easier target due to its *helix*-only structure compared to the majority of *sheet* structures in the *1URR* sample. The other columns of the table will be discussed in the next section.

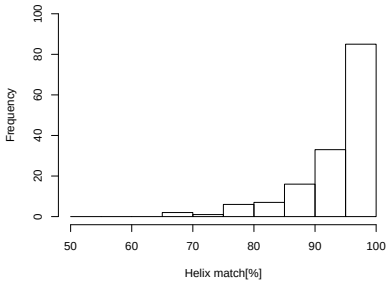
Table 12.2: Summary of tertiary structure prediction match

Protein	$\mu_{TM-Score}$	$\sigma_{TM-Score}$	$N_{TM>0.2}$	$N_{TM>0.4}$	$N_{TM>0.5}$	$N_{TM>0.6}$	$N_{TM>0.7}$	$N_{TM>0.8}$
1OAI	0.493	0.135	150	102	51	32	18	4
1URR	0.416	0.061	150	91	10	0	0	0

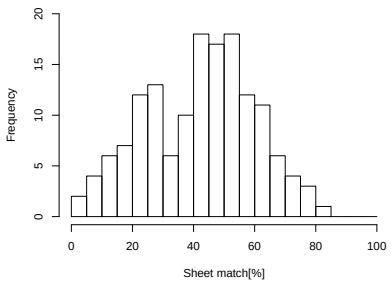
Table 12.3: Three selected generated models and their alignment scores with *1OAI* and *1URR* as reference.

Nr.	Identity	N< 5*A*	$RMSD_{N<5A}$	RMSD	GDT_{TS}	TM-Score	Helix	Sheet	Loop
1	13.6	58	1.21	1.760	92.797	0.8667	95.12	0	94.44
2	25.4	58	1.35	1.838	88.983	0.8350	95.12	0	88.89
3	18.6	56	1.84	2.722	88.136	0.8015	97.56	0	94.44

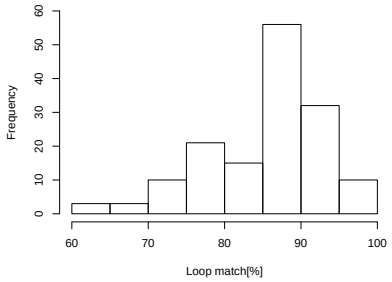
Nr.	Identity	N< 5*A*	$RMSD_{N<5A}$	RMSD	GDT_{TS}	TM-Score	Helix	Sheet	Loop
1	19.6	73	2.85	7.484	50.258	0.5374	96	75.68	71.43
2	20.6	67	3.20	4.933	50.773	0.5027	100	81.08	80
3	17.5	74	2.94	9.059	48.711	0.5138	100	72.97	80



(a) *Helix* match histogram



(b) *Sheet* match histogram



(c) *Loop* match histogram

Figure 12.3: Match histograms of *1URR* - percentage of positions in the secondary annotation of the I-TASSER predicted model that correctly matches those of the input annotation.

12.1.2 Tertiary Structure Validation Results

In the following the tertiary structure of the predicted proteins is validated by three-dimensional comparison based on the *RMSD* and *TM-Score* output in the workflow of Figure 12.1. The TM-Score detailed in [77] is a measure that is used to assess the similarity between two structures, with larger values indicating greater resemblance and 1.0 a maximum value for identical structures. According to Xu and Zhang [74] two proteins can be considered to be in the same fold if comparing them gives a TM-Score above 0.5. Though the average TM-Score is above 0.4 and close to 0.5 for the first sample, this is actually the case for $1-in-5$ for *1OAI* and $1-in-15$ for *1URR* as seen in Table 12.2. The table further shows the number N of predictions that had a TM-Score above 0.2, 0.4, 0.6, 0.7 and 0.8. The general results presented in Section 12.1.1 are confirmed here, and it is clear that the *sheet* structures of *1URR* are hard to match, and that the approach is much more successful in predicting *helix* structures (see Table 12.3).

The last step in the tertiary validation consists in superposing the fully I-TASSER

(a) *1OAI* (b) *1URR*

Figure 12.4: Secondary structure of reference (on top) compared to three selected generated models. Darker sections are *helices* lighter are *sheets* and the rest represents *loop* structure

predicted tertiary structure model of one generated sequence with the target reference. This is illustrated in Figure 12.5 and 12.6 where the first of the three individually generated sequences in Table 12.3 and Figure 12.4 is used. The models for *1OAI* are all very close to the reference seen from the high *helix* and *loop* match percentage, and in addition the first model for *1OAI* has a very low sequence identity and at the same time very high TM and GDT scores (see Table 12.3). The first model for *1URR* also has very high *helix* match percentage and good *loop* and *sheet* percentages. However the TM and GDT scores are less satisfactory. This result is visible in Fig 12.6 where the *helices* and *sheets* cannot be fully aligned with the reference, and the fact that one *sheet* has been bound to the structure in the wrong location (at the top of the figure rather than at the bottom).

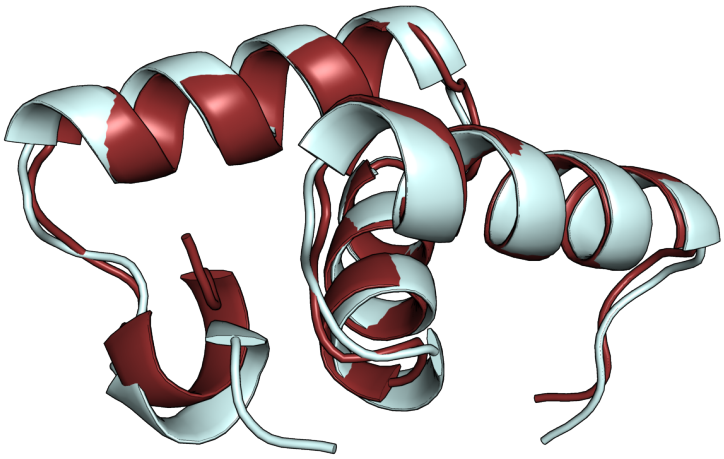


Figure 12.5: Super-positioning of a predicted model (dark red) with 1OAI reference (light blue)

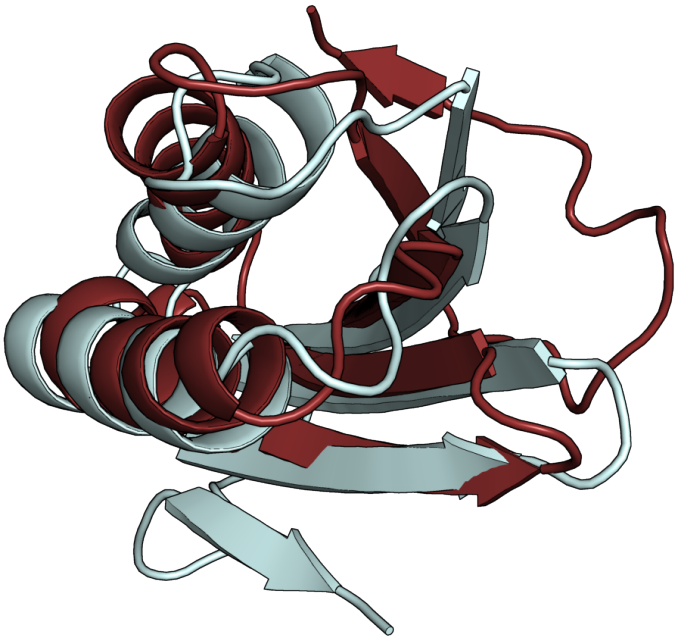


Figure 12.6: Super-positioning of a predicted model (dark red) with 1URR reference (light blue)

In Table 12.3 the second column shows sequence identity with gaps, the third shows the length of the longest continuous segment $N < 5A$ that can be fitted below a $5A$ threshold after super-positioning the two structures.

12.2 Conclusions

The secondary and tertiary validation results in this chapter demonstrate that meaningful sequences can indeed be generated from the model framework proposed. This even though the execution is fast compared to more precise methods and considering that accumulating sources of error exist in both model and test prediction. Of annotated secondary positions in the 300 test predicted sequences, *helices* are matched with above 90%, *sheets* with 42% and loops with well above 80%. When super-positioning the test predicted structures, both protein samples had many candidates of good quality $1-in-5$ for *1OAI* and $1-in-15$ for *1URR* indicated by a $TM-Score$ above 0.5.

These result indicate that the method's strength lies in smaller proteins dominated by helix regions, as was expected. Nevertheless the 42% of matching *sheet* positions is encouraging since the organisation of sheets in the actual structure is not predicted by the optimisation model. As a reminder, the objective set is efficient parallel generation of many meaningful to good solutions with high diversity, which was achieved. These results could then undergo further selection and refinement to become high quality solution instances. Otherwise the solutions could serve as the basis for further studies of the IFP solution space that may even help understanding and solving the general folding problem better.

Part IV

Conclusions and Summary

13

Conclusions

13.1 Overview

In this thesis an evolutionary-based approach to find a large amount of protein sequences that may result in a given reference secondary and tertiary structure was presented. This problem, referred to as the Inverse Folding Problem (IFP), has received a lot of attention in theoretical chemistry and biophysics over the last 30 years, mostly for its potential application in protein design. It is also of interest to study the extent of the sequence space that may produce similar tertiary structures, and how far from the original reference sequence such solutions can be found.
By defining the task as finding highly diverse sequences with most similar secondary structures, an optimisation problem was modelled to find many well-scoring sequences faster than the state-of-the-art. Thorough analysis of the optimisation problem lead to the formulation of new benchmarks based on the NK Model. This benchmark suite has been designed to share similar properties in terms of landscape analysis and algorithm behaviour.
Two algorithms were designed specifically for the problem and for maintaining high degree of diversity. The Diversity-As-Objective with Quantile Constraint (DAO-QC) NSGA-II employs *multiobjectivisation* to define diversity as additional objective which can be biased by the QC setting. The Preference Based Genetic Algorithm (PBGA) turns population fitness and diversity levels into a predefined preference that applies to the population as a whole. Both algorithms achieve superior fitness while at the same time preserving a high level of diversity compared to other methods. Finally, a large set of solutions that had been generated by the genetic algorithms were run through a leading structure prediction software. Considering the uncertainty at many levels of such a test, up to $1-in-5$ of generated sequences showed a similar folded structure to the target proteins one.

13.2 Model and Analysis

A new benchmark problem was introduced based on the *nominal discrete NKL Model* based on the well know *NK Model*. Setting $L = 20$ allows the model to work with amino-acid like sequences similar to RNA with the ultimate goal of mimicking the Inverse Folding Problem (IFP). By incorporating a sub-sampling setting, $4 \leq P \leq 20$, the internal links of the NK model can be biased and reduced in strength. Thorough problem analysis was conducted through adaptive- and random walks in terms features like fitter neighbors, auto-correlation among others as well as an extended epistatic linkage sampling around local optima. Very similar characteristics within an upper bound of a factor two were achieved in almost all tests when comparing the *NKL Model* instances to the IFP. With the sub-sampling mechanism, the landscape analysis metrics can be fitted precisely, which is advantageous when targeting different IFP samples with varying lengths and characteristics. Running the algorithms designed with different diversity maintaining features also show very similar convergence behaviour in diversity and fitness for the proposed NK models and the IFP. The algorithms tested all started to converge at approximately the same amount of evaluations, and many patterns in fitness and diversity convergence were captured by the *NKL Model*. Though the same algorithm run on the *NKL Model* would produce slightly less population diversity than the real IFP, it was found that the best performance would always be achieved by the approach that maintained a diversity of just below 40%. Hence, in this regard, the *NK Model* can be said to capture another important characteristic of the IFP, that of simulating what level of diversity produces the optimal performance. Furthermore the statistical nature of the *NK Model* with existing proofs and lemmas may provide the ground for a theoretical estimate on the number of protein sequences which fold into a given protein structure.

13.3 Algorithm Design and Experiments

Two algorithms have been developed specifically to achieve high diversity as required by the problem statement. The first algorithm addresses the requirement by extending the problem with an additional objective. Through the *multiobjectivisation* technique, the problem becomes Multi-Objective with Diversity-As-Objective (DAO). Combining the Quantile Constraint (QC) with the DAO approach allowed to shift focus arbitrarily between diversity and fitness and final results were found to be significantly better than the standard generational GA featuring removal of doubles (rdGA). With the average Hamming diversity objective and Quantile Constraint set to $C_q = 25$, the better results were achieved with statistical significance in terms of average and best fitness as well as diversity at the same time on all four protein instances.
The second algorithm developed, a novel Preference-Based Genetic Algorithm (PBGA), was presented in combination with a weighted sum model, which allowed to shift focus arbitrarily between diversity and fitness with a direct effect on the population as a whole. The PBGA was evaluated under the same conditions as the DAO-QC

and likewise achieved superior performance compared to the rdGA. A study directly comparing the PBGA to the DAO-QC showed that the PBGA is better or similar in all instances in terms of fitness, while simultaneously maintaining higher diversity. Where the DAO-QC approach can only reduce diversity when it is generated in abundance with the average Hamming diversity objective, the advantage of the PBGA is that it can gradually shift between both extremes. Hence, it can satisfy the required exploration-exploitation ratio to tackle any optimisation problem. The effect of the diversity objective seems to be sensitive to problem instance length. One could speculate that it is due to the granularity of the objective. The Hamming-distance, can take on only discrete values between 0 and N. In the extreme case with length, say 2, and a population size of 4, this means that the average diversity, $F_{div}()$, can take on $1, 2/3, 1/3, 0$. The PBGA does not experience such issues and shows more consistency in that repeated experiments for a given setting produces similar performance in terms of fitness and diversity.

13.4 IFP Result Validation

To validate, the 300 of the best generated sequences were selected and their folded structure predicted by I-TASSER, an established structure prediction software. The obtained tertiary structures were annotated by DSSP for secondary structure analysis of *Helix*, *Sheet* and *Loop* formations. As could be expected, the method works better for the sample with more defined *helical* secondary structure, and less well in *sheet* and *loop* regions, especially as the latter region is not expressed by the objective function. Indeed *sheet* formations require the tertiary structure to fold properly to be captured in secondary structure. Nevertheless the *1URR* sample *sheet* match percentage is slightly below 50% averaged over all generated predictions. In addition the majority of match-percentages are above 80% for *loops* and above 90% for *helices* in both samples. Tertiary structure validation was done by comparing the predicted structures to their respective reference by tertiary structure super-position. For both samples meaningful predictions were generated with a $TM-Score$ above 0.5 observed $1-in-5$ for *1OAI* and $1-in-15$ for *1URR*. These results indicate that this approach is able to generate a massive amount of diverse sequences, with a significant portion being likely to actually fold as the target. At the same time, the limits in terms of achieving larger formations of *sheets* are evident, though considering the optimisation model's disregard of tertiary structure, this is what could be expected. In this light the *sheet* match percentage of the *1URR* sample is encouraging.

13.5 Future Perspectives in Protein Research

Future works could address the identification of sequences that actually fold into the reference structure by designing new objectives and constraints also addressing *coil* and *beta-sheet* regions. The set of sequences that have been generated can already be

used as a starting point for other exact protein design methods. Starting from these diverse sequences it may be possible to successfully design structures close to the target, but with a very low sequence identity. In the same mindset, the process could be modified to switch objective function to a more precise model based on force field simulation functions once the algorithm has converged with the simpler objective. Another potential usage is for generating large, but meaningful, decoy sets for other studies or for finding bridges in sequence space between known proteins of the same structural classes. Finally, this study and the numerous solutions found to the IFP may help gaining new insights in the Protein Folding Prediction problem. This was one of the original ideas behind the IFP proposition [18] - to solve a simplified problem instance as a step on the path towards solving the full *ab-initio* structure prediction problem.

13.6 Future Algorithm Development

The PBGA seems to have a lot of potential for further developments. The current use of the WSM for evaluating preference is intended as a proof of concept, which could very well be extended to implement other mechanisms such as elitism, niching or adaptation and maybe even be replaced by a rule based approach using Fuzzy sets. As the behaviour of the algorithm depends on the problem instance a potential improvement lies in adaptive tuning of the preference weights. A pre-set diversity constant or progression could be targeted efficiently with the PBGA. The ensemble of the results are encouraging to proceed in developing a better algorithm as well as a better optimisation problem model.

Part V

Appendices

Bibliography

[1] E. Alba and B. Dorronsoro. The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. *IEEE Trans. Evolutionary Computation*, 9(2):126–142, 2005.

[2] Enrique Alba and Marco Tomassini. Parallelism and evolutionary algorithms. *Evolutionary Computation, IEEE Transactions on*, 6(5):443–462, 2002.

[3] Lourdes Araujo and Juan Julián Merelo. Diversity through multiculturality: Assessing migrant choice policies in an island model. *Evolutionary Computation, IEEE Transactions on*, 15(4):456–469, 2011.

[4] David Beasley, David R Bull, and Ralph R Martin. A sequential niche technique for multimodal function optimization. *Evolutionary computation*, 1(2):101–125, 1993.

[5] ML Bellows, HK Fung, MS Taylor, CA Floudas, A Lopez de Victoria, and D Morikis. New compstatin variants through two de novo protein design frameworks. *Biophysical journal*, 98(10):2337–2346, 2010.

[6] ML Bellows, MS Taylor, PA Cole, L Shen, RF Siliciano, HK Fung, and CA Floudas. Discovery of entry inhibitors for hiv-1 via a new de novo protein design framework. *Biophysical journal*, 99(10):3445–3453, 2010.

[7] A.D. Bethke. *Genetic algorithms as function optimizers*. PhD thesis.

[8] Hans-Georg Beyer. On the "explorative power" of es/ep-like algorithms. In *Evolutionary programming VII*, pages 323–334. Springer, 1998.

[9] J.U. Bowie, R. Lüthy, and D. Eisenberg. A method to identify protein sequences that fold into a known three-dimensional structure. *Science (New York, N.Y.)*, 253(5016):164–170, July 1991.

[10] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus. Charmm - a program for macromolecular energy, minimization, and dynamics calculations. *J. Comp. Chem.*, 4(2):187–217, 1983.

[11] Wilfred Chen, Fredi Brühlmann, Richard D Richins, and Ashok Mulchandani. Engineering of improved microbes and enzymes for bioremediation. *Current opinion in biotechnology*, 10(2):137–141, 1999.

[12] Helen G Cobb and John J Grefenstette. Genetic algorithms for tracking changing environments. Technical report, DTIC Document, 1993.

[13] Wendy D Cornell, Piotr Cieplak, Christopher I Bayly, Ian R Gould, Kenneth M Merz, David M Ferguson, David C Spellmeyer, Thomas Fox, James W Caldwell, and Peter A Kollman. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *Journal of the American Chemical Society*, 117(19):5179–5197, 1995.

[14] Guillaume Corriveau, Raynald Guilbault, Antoine Tahan, and Robert Sabourin. Review and study of genotypic diversity measures for real-coded representations. *Evolutionary Computation, IEEE Transactions on*, 16(5):695–710, 2012.

[15] K. A. De Jong. Analysis of the behavior of a class of genetic adaptive systems. 1975.

[16] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[17] K. Deb and A. Saha. Finding multiple solutions for multimodal optimization problems using a multi-objective evolutionary approach. In *Proceedings of the 12th annual conference on genetic and evolutionary computation*, pages 447–454. ACM, 2010.

[18] K.E. Drexler. Molecular engineering: An approach to the development of general capabilities for molecular manipulation. *Proceedings of the National Academy of Sciences*, 78(9):5275–5278, 1981.

[19] Agoston E Eiben and CA Schippers. On evolutionary exploration and exploitation. *Fundamenta Informaticae*, 35(1-4):35–50, 1998.

[20] Marco Farina, Kalyanmoy Deb, and Paolo Amato. Dynamic multiobjective optimization problems: test cases, approximations, and applications. *Evolutionary Computation, IEEE Transactions on*, 8(5):425–442, 2004.

[21] Lawrence J Fogel, Alvin J Owens, and Michael J Walsh. Artificial intelligence through simulated evolution. 1966.

[22] Ho Ki Fung, Christodoulos A Floudas, Martin S Taylor, Li Zhang, and Dimitrios Morikis. Toward full-sequence de novo protein design with flexible templates for human beta-defensin-2. *Biophysical journal*, 94(2):584–599, 2008.

[23] David E Goldberg. Genetic algorithms and walsh functions: Part ii , deception and its analysis. *Complex Systems*, 3:153–171, 1989.

[24] David E Goldberg, Kalyanmoy Deb, and Je rey Horn. Massive multimodality, deception, and genetic algorithms. *Urbana*, 51:61801, 1992.

[25] D.E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*, pages 41–49, 1987.

[26] Peter Güntert. Automated nmr structure calculation with cyana. In *Protein NMR Techniques*, pages 353–378. Springer, 2004.

[27] B. Gutte, M. Däumigen, and E.A. Wittschieber. Design, synthesis and characterisation of a 34-residue polypeptide that interacts with nucleic acids. *Nature*, 281(5733):650–655, 1979.

[28] Pehr B Harbury, Joseph J Plecs, Bruce Tidor, Tom Alber, and Peter S Kim. High-resolution protein design with backbone freedom. *Science*, 282(5393):1462–1467, 1998.

[29] John H Holland. Genetic algorithms and the optimal allocation of trials. *SIAM Journal on Computing*, 2(2):88–105, 1973.

[30] Jianjun Hu and Erik Goodman. Robust and efficient genetic algorithms with hierarchical niching and a sustainable evolutionary computation model. In *Genetic and Evolutionary Computation–GECCO 2004*, pages 1220–1232. Springer, 2004.

[31] Y Isogai, M Ota, T Fujisawa, H Izuno, M Mukai, H Nakamura, T Iizuka, and K Nishikawa. Design and synthesis of a globin fold. *Biochemistry*, 38(23):7431—7443, June 1999.

[32] Mikkel T Jensen. Guiding single-objective optimization using multi-objective methods. In *Applications of Evolutionary Computing*, pages 268–279. Springer, 2003.

[33] D.T. Jones. De *novo* protein design using pairwise potentials and a genetic algorithm. *Protein Science*, 3:567–574, 1994.

[34] W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–637, December 1983.

[35] Stuart A. Kauffman. *The origins of order: Self-organization and selection in evolution*. Oxford university press, 1993.

[36] Stuart A Kauffman and Edward D Weinberger. The nk model of rugged fitness landscapes and its application to maturation of the immune response. *Journal of theoretical biology*, 141(2):211–245, 1989.

[37] F. Klein, H. Mouquet, P. Dosenovic, J.F. Scheid, L. Scharf, and M.C. Nussenzweig. Antibodies in HIV-1 vaccine development and therapy. *Science (New York, N.Y.)*, 341(6151):1199–204, September 2013.

[38] J.L. Klepeis, C.A. Floudas, D. Morikis, C.G. Tsokos, and J.D. Lambris. Design of peptide analogues with improved activity using a novel de novo protein design approach. *Industrial & engineering chemistry research*, 43(14):3817–3826, 2004.

[39] Joshua D Knowles, Richard A Watson, and David W Corne. Reducing local optima in single-objective problems by multi-objectivization. In *Evolutionary multi-criterion optimization*, pages 269–283. Springer, 2001.

[40] Brian Kuhlman and David Baker. Native protein sequences are close to optimal for their structures. *Proceedings of the National Academy of Sciences*, 97(19):10383–10388, 2000.

[41] J.L.J. Laredo, S.S. Nielsen, G. Danoy, P. Bouvry, and C.M. Fernandes. Cooperative selection: Improving tournament selection via altruism. *accepted for publication in EvoCOP14 - 14th European Conference on Evolutionary Computation in Combinatorial Optimisation*, 2014.

[42] Rui Li, Michael TM Emmerich, Jeroen Eggermont, Ernst GP Bovenkamp, Thomas Bäck, Jouke Dijkstra, and Johan HC Reiber. Mixed-integer nk landscapes. In *Parallel Problem Solving from Nature-PPSN IX*, pages 42–51. Springer, 2006.

[43] Samir W Mahfoud. Crowding and preselection revisited. *Urbana*, 51:61801, 1992.

[44] Samir W Mahfoud. A comparison of parallel and sequential niching methods. In *Conference on genetic algorithms*, volume 136, page 143. Citeseer, 1995.

[45] Melanie Mitchell, Stephanie Forrest, and John H Holland. The royal road for genetic algorithms: Fitness landscapes and ga performance. *Ann Arbor*, 1001:48109.

[46] P. Mitra, D. Shultis, J.R. Brender, J. Czajka, D. Marsh, F. Gray, T. Cierpicki, and Y. Zhang. An evolution-based approach to de novo protein design and case study on mycobacterium tuberculosis. *PLoS computational biology*, 9(10):e1003298, 2013.

[47] Ronald W Morrison and Kenneth A De Jong. Measurement of population diversity. In *Artificial Evolution*, pages 31–41. Springer, 2002.

[48] John Moult, Krzysztof Fidelis, Andriy Kryshtafovych, Torsten Schwede, and Anna Tramontano. Critical assessment of methods of protein structure prediction (casp)—round x. *Proteins: Structure, Function, and Bioinformatics*, 82(S2):1–6, 2014.

[49] John Moult, Jan T Pedersen, Richard Judson, and Krzysztof Fidelis. A large-scale experiment to assess protein structure prediction methods. *Proteins: Structure, Function, and Bioinformatics*, 23(3):ii–iv, 1995.

[50] Trung Thanh Nguyen, Shengxiang Yang, and Juergen Branke. Evolutionary dynamic optimization: A survey of the state of the art. *Swarm and Evolutionary Computation*, 6:1–24, 2012.

[51] Sune S Nielsen, Grégoire Danoy, Wiktor Jurkowski, Juan Luis Jiménez Laredo, Reinhard Schneider, El-Ghazali Talbi, and Pascal Bouvry. A novel multi-objectivisation approach for optimising the protein inverse folding problem. In *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, page (To appear). Springer, 2015.

[52] C. Pabo. Molecular technology. Designing proteins and peptides. *Nature*, 301(5897), January 1983.

[53] Alain Pétrowski. A clearing procedure as a niching method for genetic algorithms. In *International conference on evolutionary computation*, pages 798–803, 1996.

[54] Jay W Ponder et al. Tinker: Software tools for molecular design. *Washington University School of Medicine, Saint Louis, MO*, 3, 2004.

[55] Jay W. Ponder and Frederic M. Richards. Tertiary templates for proteins: Use of packing criteria in the enumeration of allowed sequences for different structural classes. *Journal of Molecular Biology*, 193(4):775 – 791, 1987.

[56] Ingo Rechenberg. Cybernetic solution path of an experimental problem. 1965.

[57] Ingo Rechenberg. Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution. 1973.

[58] Carol A Rohl, Charlie EM Strauss, Kira MS Misura, and David Baker. Protein structure prediction using rosetta. *Methods in enzymology*, 383:66–93, 2004.

[59] B. Rost and C. Sander. Combining evolutionary information and neural networks to predict protein secondary structure. *Proteins*, 19(1):55–72, May 1994.

[60] H. Shimodaira. Dcga: A diversity control oriented genetic algorithm. In *ICTAI*, pages 367–374, 1997.

[61] J. Smadbeck, M.B. Peterson, G.A. Khoury, M.S. Taylor, and C.A. Floudas. Protein wisdom: a workbench for in silico de novo design of biomolecules. *J Vis Exp*, 2013.

[62] A. Su and S.L. Mayo. Coupling backbone flexibility and amino acid sequence selection in protein design. *Protein Science*, 6(8):1701–1707, 1997.

[63] El-Ghazali Talbi. *Metaheuristics: From Design to Implementation*. Wiley, 2009.

[64] A. Toffolo and E. Benini. Genetic diversity as an objective in multi-objective evolutionary algorithms. *Evol. Comput.*, 11(2):151–167, May 2003.

[65] Yasuhiro Tsujimura and Mitsuo Gen. Entropy-based genetic algorithm for solving tsp. In *Knowledge-Based Intelligent Electronic Systems, 1998. Proceedings KES'98. 1998 Second International Conference on*, volume 2, pages 285–290. IEEE, 1998.

[66] S. Varrette, P. Bouvry, H. Cartiaux, and F. Georgatos. Management of an academic hpc cluster: The ul experience. In *Proc. of the 2014 Intl. Conf. on High Performance Computing & Simulation (HPCS 2014)*, Bologna, Italy, July 2014. IEEE.

[67] Javier E Vitela and Octavio Castaños. A real-coded niching memetic algorithm for continuous multimodal function optimization. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 2170–2177. IEEE, 2008.

[68] Christopher A. Voigt, Stephen L. Mayo, Frances H. Arnold, and Zhen-Gang Wang. Computational method to reduce the search space for directed protein evolution. *Proceedings of the National Academy of Sciences of the United States of America*, 98(7):3778–3783, March 2001.

[69] L. Wernisch, S. Hery, and S. Wodak. Automatic protein design with all atom force-fields by exact and heuristic optimization. *Journal of Molecular Biology*, 301(3):713–736, 2000.

[70] S. Wessing, M. Preuss, and G. Rudolph. Niching by multiobjectivization with neighbor information: Trade-offs and benefits. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 103–110. IEEE, 2013.

[71] Darrell Whitley, Soraya Rana, and Robert B Heckendorn. The island model genetic algorithm: On separability, population size and convergence. *Journal of Computing and Information Technology*, 7:33–48, 1999.

[72] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[73] David H. Wolpert and William G. Macready. No free lunch theorems for search, 1995.

[74] Jinrui Xu and Yang Zhang. How significant is a protein structure similarity with tm-score= 0.5? *Bioinformatics*, 26(7):889–895, 2010.

[75] Jianyi Yang, Renxiang Yan, Ambrish Roy, Dong Xu, Jonathan Poisson, and Yang Zhang. The i-tasser suite: protein structure and function prediction. *Nature methods*, 12(1):7–8, 2015.

[76] A. Zemla. LGA: A method for finding 3D similarities in protein structures. *Nucleic acids research*, 31(13):3370–3374, July 2003.

[77] Yang Zhang and Jeffrey Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710, 2004.

[78] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.

Publications

[1] Sune S. Nielsen, Grégoire Danoy, Wiktor Jurkowski, Roland Krause, Reinhard Schneider, El-Ghazali Talbi and Pascal Bouvry. *Evolutionary Algorithms for the Inverse Protein Folding Problem*. to appear in Handbook of Heuristics, Springer

[2] Sune S. Nielsen, Christof Ferreira Torres, Grégoire Danoy, Pascal Bouvry. *Tackling the IFP Problem with the Preference-Based Genetic Algorithm* GECCO 2016 Proceedings: 965-972

[3] Sune S. Nielsen, Grégoire Danoy, Wiktor Jurkowski, Juan Luis Jiménez Laredo, Reinhard Schneider, El-Ghazali Talbi, Pascal Bouvry. *A Novel Multi-objectivisation Approach for Optimising the Protein Inverse Folding Problem* EvoApplications 2015: 14-25

[4] Christof Ferreira Torres, Sune S. Nielsen, Grégoire Danoy, Pascal Bouvry, El-Ghazali Talbi. *Preference-Based Genetic Algorithm for Solving the Bio-Inspired NK Landscape Benchmark.* 7th European Symposium on Computational Intelligence and Mathematics (ESCIM) 2015

[5] Sune S. Nielsen, Grégoire Danoy, Pascal Bouvry, El-Ghazali Talbi. *NK Landscape Instances Mimicking the Protein Inverse Folding Problem Towards Future Benchmarks.* GECCO (Companion) 2015: 915-921

[6] Juan Luis Jiménez Laredo, Sune S. Nielsen, Grégoire Danoy, Pascal Bouvry, Carlos M. Fernandes. *Cooperative Selection: Improving Tournament Selection via Altruism.* EvoCOP 2014: 85-96

[7] Sune S. Nielsen, Grégoire Danoy, Pascal Bouvry. *Vehicular mobility model optimization using cooperative coevolutionary genetic algorithms* GECCO 2013: 1349-1356

[8] Sune S. Nielsen, Bernabé Dorronsoro, Grégoire Danoy, Pascal Bouvry. *Novel efficient asynchronous cooperative co-evolutionary multi-objective algorithms.* IEEE Congress on Evolutionary Computation 2012: 1-7

List of Figures

List of Tables

Appendix A

Detailed Results on Protein Samples

A.1 Description

The following four sections each contain one of four studies. The *Baseline study* in Section A.2 compares selected existing genetic algorithms, the *Diversity as Objective with Quantile Constraint study* in Section A.3 and *Preference Based Genetic Algorithm study* in Section A.4 test the two developed algorithms with different settings. Section A.5 *Summary of Algorithm Experiments* summarises and compares both developed algorithms with a chosen baseline. Each study displays the same cross-comparison of final results and convergence plots for each sample. The metrics studied are *Average Fitness*, *Best Fitness* and *Diversity* of 30 individual runs per algorithm and setting presented. For more details on the experiment setup, please refer to Chapter 9.

A.2 Baseline Study

	rdGA	gGA	scGA	ssGA
rdGA	⁄	−0.0144▽	0.0126▲	0.00434 -
gGA		⁄	0.027▲	0.0187▲
scGA			⁄	−0.00826 -
ssGA				⁄

Table A.1: Base study convergence statistics for 1B3A Cross comparison Average fitness

	rdGA	gGA	scGA	ssGA
rdGA	⁄	−0.0114▽	−0.015▽	−0.0232▽
gGA		⁄	−0.00361 -	−0.0119▽
scGA			⁄	−0.00826 -
ssGA				⁄

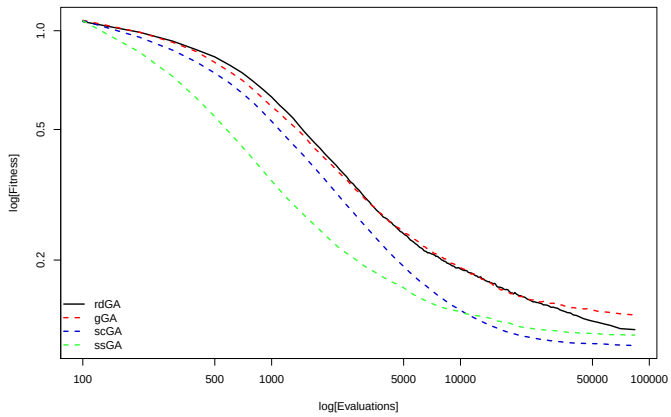
Table A.2: Base study convergence statistics for 1B3A Cross comparison Best fitness

	rdGA	gGA	scGA	ssGA
rdGA	⁄	18.800▽	31.587▽	31.783▽
gGA		⁄	12.787▽	12.983▽
scGA			⁄	0.196▽
ssGA				⁄

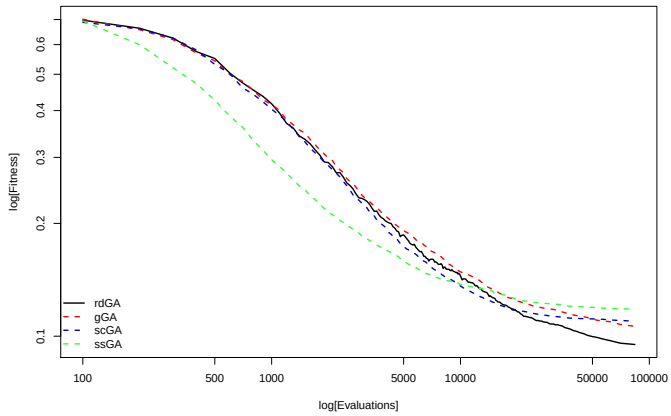
Table A.3: Base study convergence statistics for 1B3A Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.122_{\pm 1.14e-02}$	$0.0948_{\pm 9.42e-03}$	$31.886_{\pm 3.20e+00}$
gGA	$0.137_{\pm 1.46e-02}$	$0.106_{\pm 1.70e-02}$	$13.086_{\pm 2.11e+00}$
scGA	$0.110_{\pm 2.10e-02}$	$0.110_{\pm 2.10e-02}$	$0.299_{\pm 4.59e-01}$
ssGA	$0.118_{\pm 2.16e-02}$	$0.118_{\pm 2.17e-02}$	$0.103_{\pm 2.98e-01}$

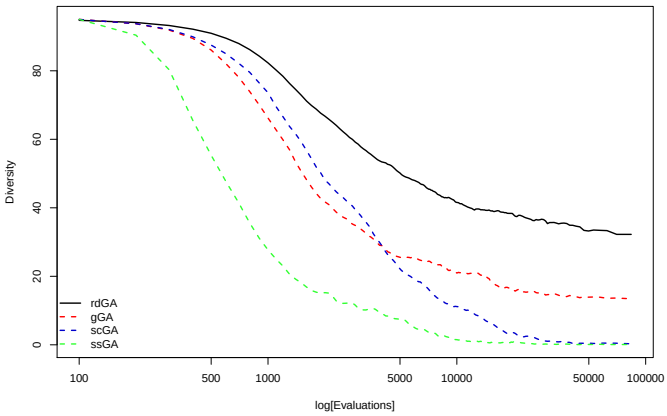
Table A.4: Base study convergence statistics for 1B3A Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure A.1: Convergence analysis base study *1B3A*

	rdGA	gGA	scGA	ssGA
rdGA	⁄	$-0.0101\triangledown$	$3e-04$ -	$-0.0049\triangledown$
gGA		⁄	$0.0104\blacktriangle$	0.00516 -
scGA			⁄	$-0.0052\triangledown$
ssGA				⁄

Table A.5: Base study convergence statistics for 256B Cross comparison Average fitness

	rdGA	gGA	scGA	ssGA
rdGA	⁄	$-0.00893\triangledown$	$-0.0105\triangledown$	$-0.0158\triangledown$
gGA		⁄	-0.00153 -	$-0.00682\triangledown$
scGA			⁄	$-0.0053\triangledown$
ssGA				⁄

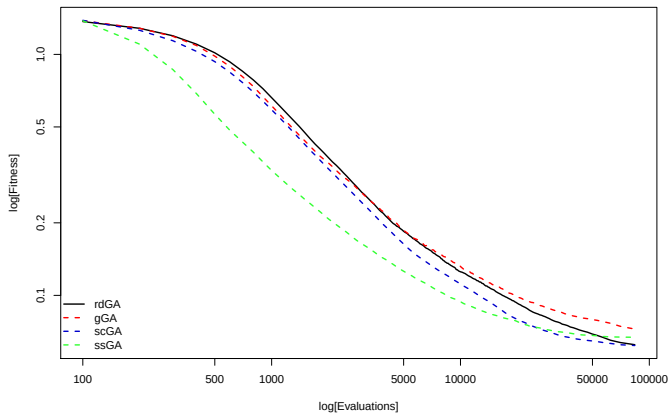
Table A.6: Base study convergence statistics for 256B Cross comparison Best fitness

	rdGA	gGA	scGA	ssGA
rdGA	⁄	$11.531\triangledown$	$19.278\triangledown$	$19.792\triangledown$
gGA		⁄	$7.748\triangledown$	$8.261\triangledown$
scGA			⁄	$0.514\triangledown$
ssGA				⁄

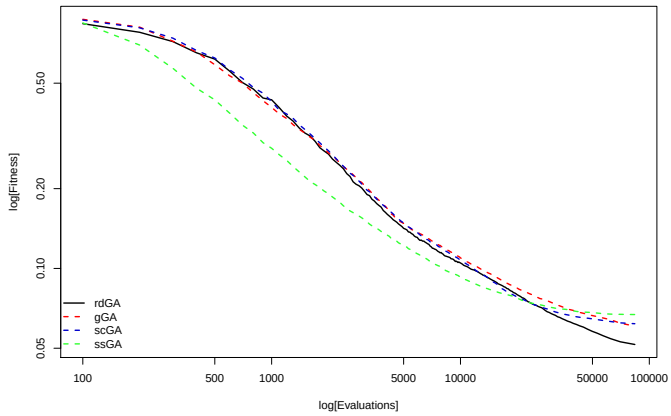
Table A.7: Base study convergence statistics for 256B Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.0621_{\pm 5.50e-03}$	$0.0512_{\pm 5.94e-03}$	$19.815_{\pm 2.26e+00}$
gGA	$0.0721_{\pm 6.41e-03}$	$0.0601_{\pm 5.35e-03}$	$8.285_{\pm 1.20e+00}$
scGA	$0.0618_{\pm 6.58e-03}$	$0.0617_{\pm 6.59e-03}$	$0.537_{\pm 6.08e-01}$
ssGA	$0.067_{\pm 1.01e-02}$	$0.067_{\pm 1.01e-02}$	$0.0235_{\pm 3.66e-02}$

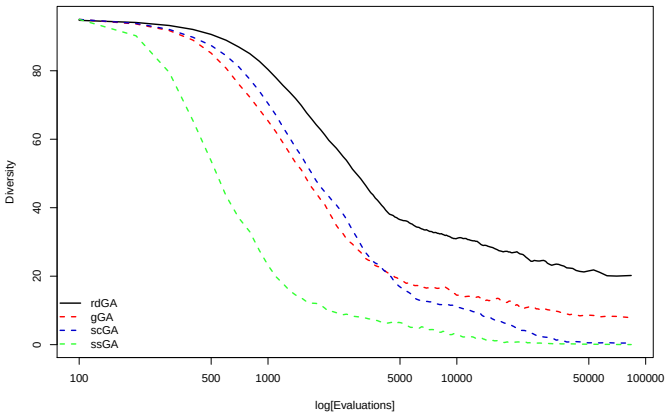
Table A.8: Base study convergence statistics for 256B Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure A.2: Convergence analysis base study *256B*

	rdGA	gGA	scGA	ssGA
rdGA	⁄	−0.00755▽	0.0081▲	0.00527▲
gGA		⁄	0.0157▲	0.0128▲
scGA			⁄	−0.00283▽
ssGA				⁄

Table A.9: Base study convergence statistics for 1OAI Cross comparison Average fitness

	rdGA	gGA	scGA	ssGA
rdGA	⁄	−0.00418▽	−0.00751▽	−0.0103▽
gGA		⁄	−0.00333 -	−0.00613▽
scGA			⁄	−0.0028▽
ssGA				⁄

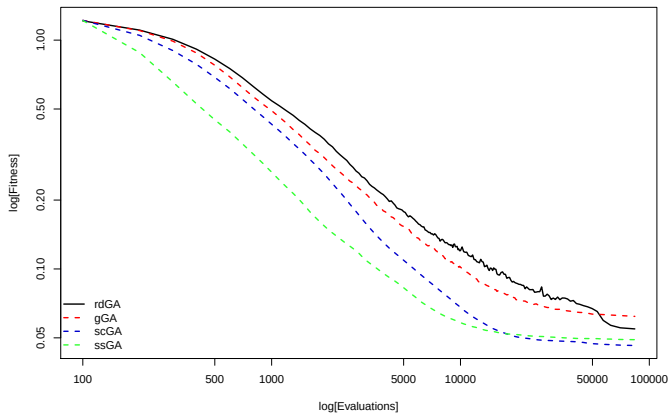
Table A.10: Base study convergence statistics for 1OAI Cross comparison Best fitness

	rdGA	gGA	scGA	ssGA
rdGA	⁄	15.649▽	28.219▽	28.819▽
gGA		⁄	12.570▽	13.170▽
scGA			⁄	0.600▽
ssGA				⁄

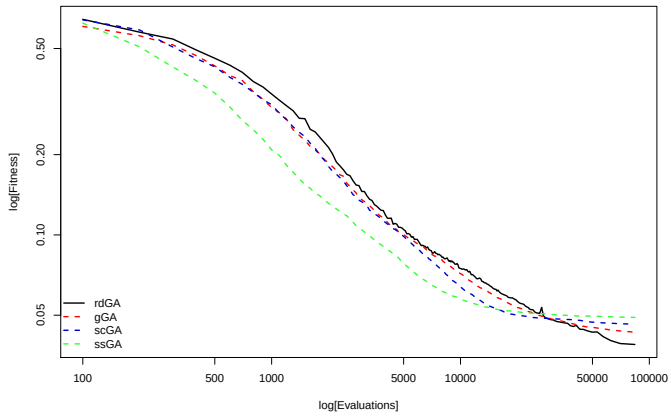
Table A.11: Base study convergence statistics for 1OAI Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.0543_{\pm 2.75e-03}$	$0.0387_{\pm 1.99e-03}$	$29.084_{\pm 2.10e+00}$
gGA	$0.0619_{\pm 5.41e-03}$	$0.0429_{\pm 3.11e-03}$	$13.434_{\pm 2.20e+00}$
scGA	$0.0462_{\pm 8.20e-03}$	$0.0462_{\pm 8.21e-03}$	$0.865_{\pm 1.02e+00}$
ssGA	$0.0491_{\pm 5.18e-03}$	$0.049_{\pm 5.20e-03}$	$0.265_{\pm 4.77e-01}$

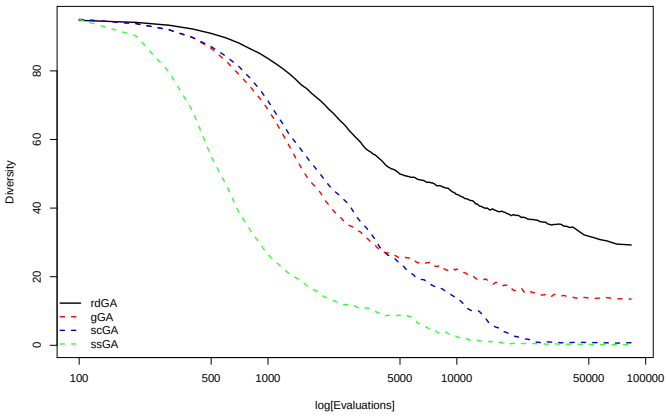
Table A.12: Base study convergence statistics for 1OAI Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure A.3: Convergence analysis base study *1OAI*

	rdGA	gGA	scGA	ssGA
rdGA	⁄	−0.00875▽	0.0122▲	0.00424▲
gGA		⁄	0.021▲	0.013▲
scGA			⁄	−0.00797▽
ssGA				⁄

Table A.13: Base study convergence statistics for 1URR Cross comparison Average fitness

	rdGA	gGA	scGA	ssGA
rdGA	⁄	−0.00718▽	−0.00545▽	−0.0135▽
gGA		⁄	0.00174 -	−0.00629▽
scGA			⁄	−0.00802▽
ssGA				⁄

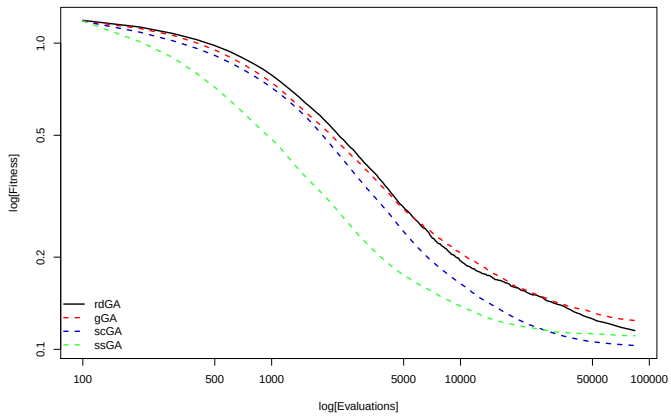
Table A.14: Base study convergence statistics for 1URR Cross comparison Best fitness

	rdGA	gGA	scGA	ssGA
rdGA	⁄	14.132▽	22.707▽	22.926▽
gGA		⁄	8.575▽	8.794▽
scGA			⁄	0.219▽
ssGA				⁄

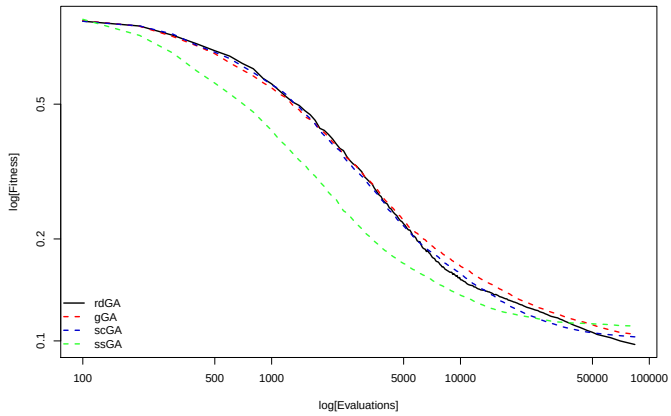
Table A.15: Base study convergence statistics for 1URR Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.115_{\pm 6.30e-03}$	$0.0971_{\pm 4.96e-03}$	$22.962_{\pm 2.32e+00}$
gGA	$0.124_{\pm 8.73e-03}$	$0.104_{\pm 5.89e-03}$	$8.830_{\pm 1.34e+00}$
scGA	$0.103_{\pm 5.92e-03}$	$0.103_{\pm 5.93e-03}$	$0.255_{\pm 4.15e-01}$
ssGA	$0.111_{\pm 8.45e-03}$	$0.111_{\pm 8.45e-03}$	$0.0358_{\pm 1.63e-01}$

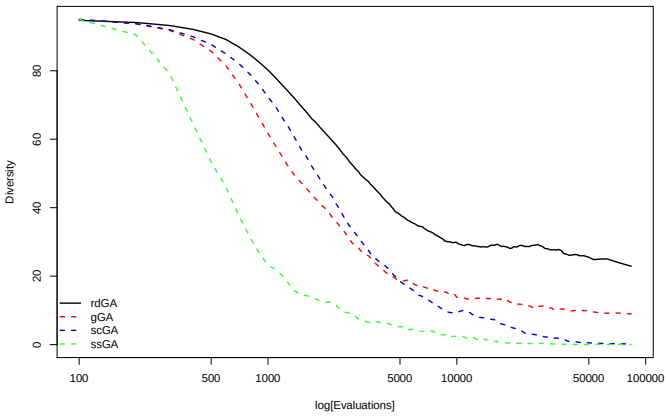
Table A.16: Base study convergence statistics for 1URR Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure A.4: Convergence analysis base study *1URR*

A.3 Diversity as Objective with Quantile Constraint Study

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−0.103▽	0.00599▲	0.0168▲	0.0221▲	0.0204▲	0.0185▲
DAOA-QC0		⁄	0.109▲	0.120▲	0.125▲	0.124▲	0.122▲
DAOA-QC10			⁄	0.0108▲	0.0161▲	0.0144▲	0.0125▲
DAOA-QC25				⁄	0.00529▲	0.00355▲	0.00165▲
DAOM-QC0					⁄	−0.00173 -	−0.00364 -
DAOM-QC10						⁄	−0.00191 -
DAOM-QC25							⁄

Table A.17: DAO-QC Convergence statistics for 1B3A Cross comparison Average fitness

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−0.0112▽	0.00522▲	0.00856▲	0.00189 -	−0.00495 -	−0.0047 -
DAOA-QC0		⁄	0.0165▲	0.0198▲	0.0131▲	0.00629▲	0.00654▲
DAOA-QC10			⁄	0.00333▲	−0.00333 -	−0.0102 -	−0.00993▽
DAOA-QC25				⁄	−0.00667▽	−0.0135▽	−0.0133▽
DAOM-QC0					⁄	−0.00684 -	−0.00659 -
DAOM-QC10						⁄	0.000249 -
DAOM-QC25							⁄

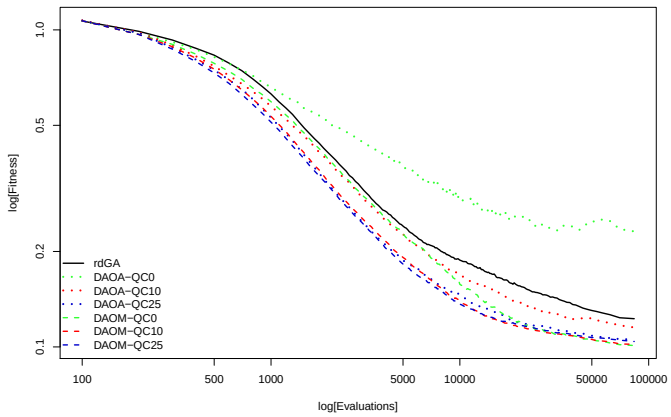
Table A.18: DAO-QC Convergence statistics for 1B3A Cross comparison Best fitness

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−47.160▲	−38.599▲	−18.534▲	22.672▽	25.954▽	25.355▽
DAOA-QC0		⁄	8.561▽	28.626▽	69.831▽	73.114▽	72.515▽
DAOA-QC10			⁄	20.065▽	61.270▽	64.553▽	63.954▽
DAOA-QC25				⁄	41.205▽	44.488▽	43.889▽
DAOM-QC0					⁄	3.282▽	2.683 -
DAOM-QC10						⁄	−0.599 -
DAOM-QC25							⁄

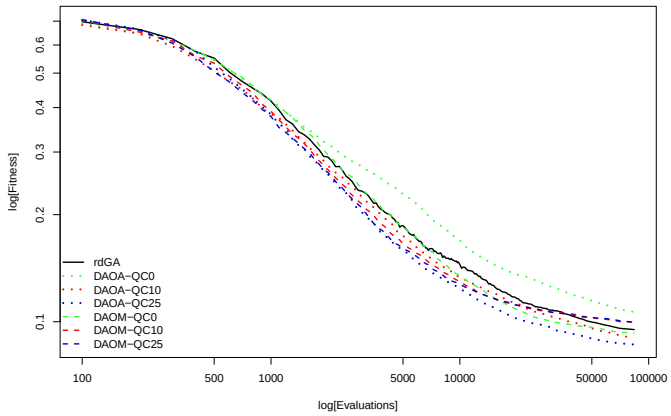
Table A.19: DAO-QC Convergence statistics for 1B3A Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.122_{\pm 1.14e-02}$	$0.0948_{\pm 9.42e-03}$	$31.886_{\pm 3.20e+00}$
DAOA-QC0	$0.226_{\pm 9.64e-02}$	$0.106_{\pm 7.75e-03}$	$79.046_{\pm 3.30e+00}$
DAOA-QC10	$0.116_{\pm 1.15e-02}$	$0.0895_{\pm 4.14e-03}$	$70.485_{\pm 4.63e+00}$
DAOA-QC25	$0.106_{\pm 7.35e-03}$	$0.0862_{\pm 4.36e-03}$	$50.420_{\pm 6.08e+00}$
DAOM-QC0	$0.100_{\pm 2.02e-02}$	$0.0929_{\pm 9.29e-03}$	$9.215_{\pm 1.17e+01}$
DAOM-QC10	$0.102_{\pm 1.99e-02}$	$0.0997_{\pm 2.00e-02}$	$5.932_{\pm 1.30e+00}$
DAOM-QC25	$0.104_{\pm 1.71e-02}$	$0.0995_{\pm 1.72e-02}$	$6.531_{\pm 1.22e+00}$

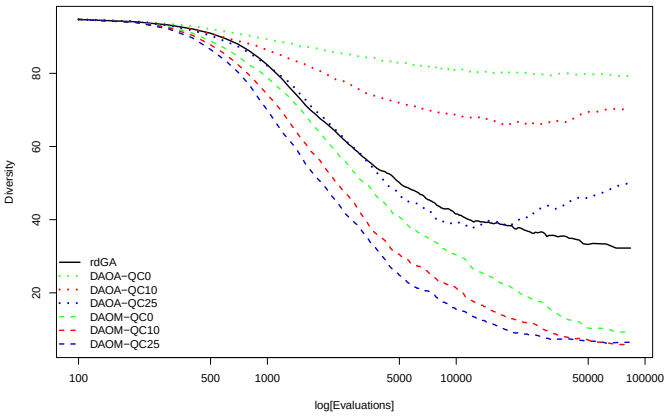
Table A.20: DAO-QC Convergence statistics for 1B3A Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure A.5: Convergence analysis DAO-QC *1B3A*

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−0.0823▽	−0.00471▽	0.00642▲	0.00617▲	0.00577▲	0.00503▲
DAOA-QC0		⁄	0.0776▲	0.0888▲	0.0885▲	0.0881▲	0.0874▲
DAOA-QC10			⁄	0.0111▲	0.0109 -	0.0105 -	0.00974 -
DAOA-QC25				⁄	−0.000241 -	−0.000646 -	−0.00139 -
DAOM-QC0					⁄	−0.000404 -	−0.00114 -
DAOM-QC10						⁄	−0.00074 -
DAOM-QC25							⁄

Table A.21: DAO-QC Convergence statistics for 256B Cross comparison Average fitness

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−0.0127▽	0.00292 -	0.00376▲	−0.00289▽	−0.00388▽	−0.00396▽
DAOA-QC0		⁄	0.0156▲	0.0164▲	0.00978▲	0.00879▲	0.00871▲
DAOA-QC10			⁄	0.000833 -	−0.00582▽	−0.00681▽	−0.00689▽
DAOA-QC25				⁄	−0.00665▽	−0.00764▽	−0.00772▽
DAOM-QC0					⁄	−0.000991 -	−0.00107 -
DAOM-QC10						⁄	$-7.85e-05$ -
DAOM-QC25							⁄

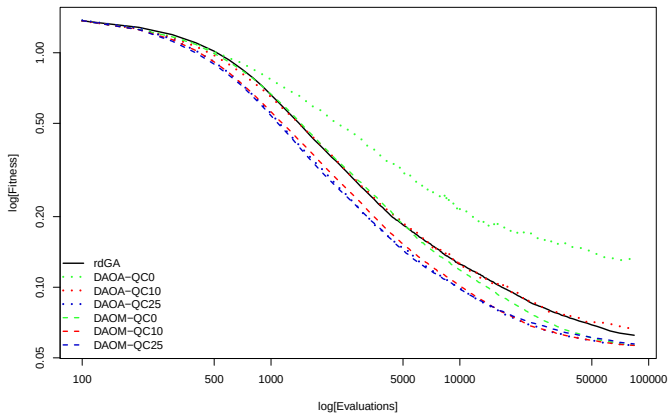
Table A.22: DAO-QC Convergence statistics for 256B Cross comparison Best fitness

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−53.236▲	−38.334▲	−16.675▲	14.326▽	15.387▽	15.485▽
DAOA-QC0		⁄	14.902▽	36.561▽	67.561▽	68.623▽	68.720▽
DAOA-QC10			⁄	21.659▽	52.659▽	53.721▽	53.818▽
DAOA-QC25				⁄	31.001▽	32.062▽	32.160▽
DAOM-QC0					⁄	1.061▽	1.159▽
DAOM-QC10						⁄	0.0976 -
DAOM-QC25							⁄

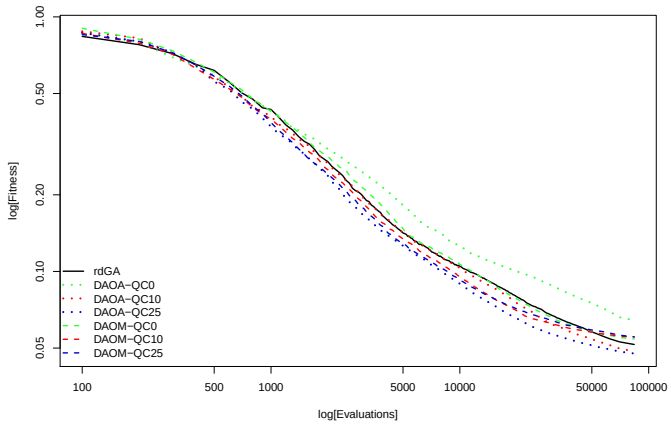
Table A.23: DAO-QC Convergence statistics for 256B Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.0621_{\pm 5.50e-03}$	$0.0512_{\pm 5.94e-03}$	$19.815_{\pm 2.26e+00}$
DAOA-QC0	$0.144_{\pm 1.11e-01}$	$0.0639_{\pm 4.06e-03}$	$73.051_{\pm 3.22e+00}$
DAOA-QC10	$0.0668_{\pm 3.91e-02}$	$0.0483_{\pm 3.26e-03}$	$58.149_{\pm 6.47e+00}$
DAOA-QC25	$0.0557_{\pm 4.20e-03}$	$0.0475_{\pm 3.82e-03}$	$36.490_{\pm 6.29e+00}$
DAOM-QC0	$0.0559_{\pm 6.76e-03}$	$0.0541_{\pm 6.51e-03}$	$5.490_{\pm 1.18e+00}$
DAOM-QC10	$0.0563_{\pm 7.73e-03}$	$0.0551_{\pm 7.55e-03}$	$4.428_{\pm 9.44e-01}$
DAOM-QC25	$0.057_{\pm 5.64e-03}$	$0.0552_{\pm 5.42e-03}$	$4.331_{\pm 7.66e-01}$

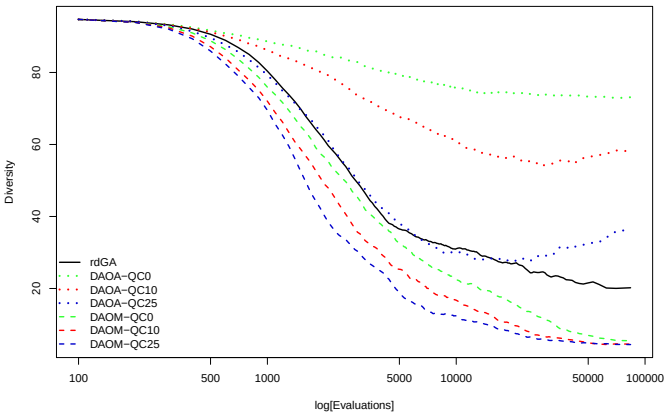
Table A.24: DAO-QC Convergence statistics for 256B Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure A.6: Convergence analysis DAO-QC *256B*

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	／	−0.121▽	−0.00781▽	0.00187▲	0.0132▲	0.0129▲	0.010▲
DAOA-QC0		／	0.113▲	0.123▲	0.134▲	0.134▲	0.131▲
DAOA-QC10			／	0.00968▲	0.021▲	0.0207▲	0.0178▲
DAOA-QC25				／	0.0114▲	0.011▲	0.00815▲
DAOM-QC0					／	−0.000345 -	−0.00322▽
DAOM-QC10						／	−0.00288▽
DAOM-QC25							／

Table A.25: DAO-QC Convergence statistics for 1OAI Cross comparison Average fitness

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	／	−0.00941▽	−0.00113▽	0.00195▲	0.00065 -	−0.00116 -	−0.00314▽
DAOA-QC0		／	0.00828▲	0.0114▲	0.0101▲	0.00825▲	0.00627▲
DAOA-QC10			／	0.00308▲	0.00178▲	$-2.81e-05$ -	−0.00201▽
DAOA-QC25				／	−0.0013▽	−0.00311▽	−0.00508▽
DAOM-QC0					／	−0.00181▽	−0.00379▽
DAOM-QC10						／	−0.00198 -
DAOM-QC25							／

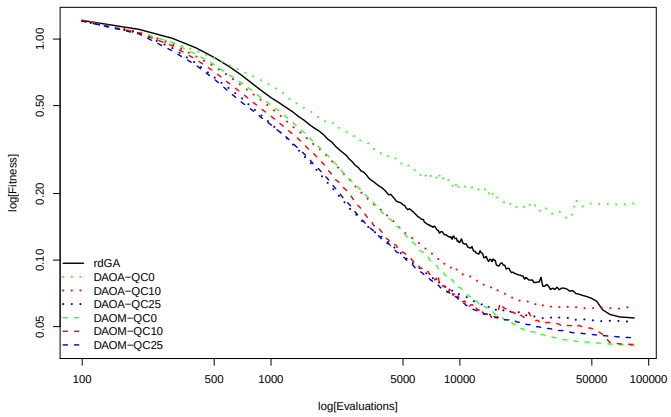
Table A.26: DAO-QC Convergence statistics for 1OAI Cross comparison Best fitness

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	／	−50.500▲	−43.389▲	−22.895▲	20.030▽	21.455▽	21.549▽
DAOA-QC0		／	7.111▽	27.605▽	70.530▽	71.955▽	72.049▽
DAOA-QC10			／	20.494▽	63.419▽	64.844▽	64.938▽
DAOA-QC25				／	42.925▽	44.350▽	44.444▽
DAOM-QC0					／	1.424▽	1.518▽
DAOM-QC10						／	0.094 -
DAOM-QC25							／

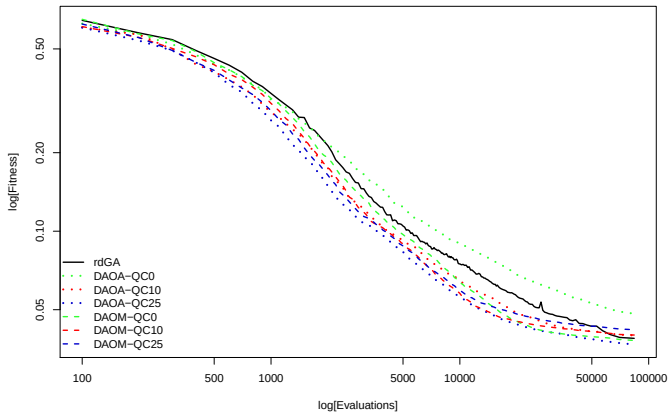
Table A.27: DAO-QC Convergence statistics for 1OAI Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.0543_{\pm 2.75e-03}$	$0.0387_{\pm 1.99e-03}$	$29.084_{\pm 2.10e+00}$
DAOA-QC0	$0.175_{\pm 1.03e-01}$	$0.0481_{\pm 2.47e-03}$	$79.584_{\pm 2.37e+00}$
DAOA-QC10	$0.0621_{\pm 6.26e-03}$	$0.0398_{\pm 2.04e-03}$	$72.473_{\pm 3.08e+00}$
DAOA-QC25	$0.0525_{\pm 3.80e-03}$	$0.0368_{\pm 1.66e-03}$	$51.979_{\pm 5.24e+00}$
DAOM-QC0	$0.0411_{\pm 3.49e-03}$	$0.0381_{\pm 2.56e-03}$	$9.054_{\pm 1.93e+00}$
DAOM-QC10	$0.0414_{\pm 3.83e-03}$	$0.0399_{\pm 3.55e-03}$	$7.629_{\pm 1.92e+00}$
DAOM-QC25	$0.0443_{\pm 4.31e-03}$	$0.0418_{\pm 3.91e-03}$	$7.535_{\pm 7.91e-01}$

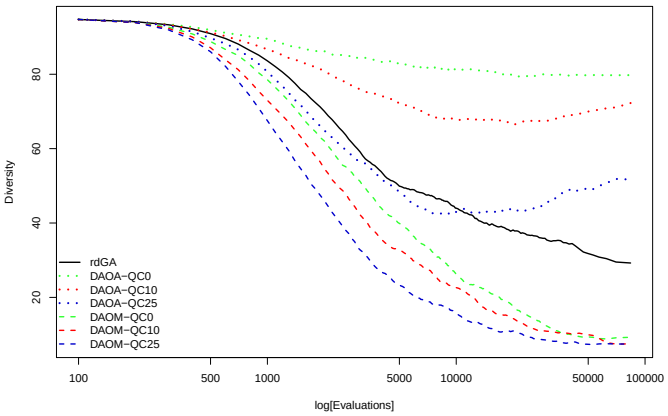
Table A.28: DAO-QC Convergence statistics for 1OAI Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure A.7: Convergence analysis DAO-QC *1OAI*

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−0.103▽	0.00791▲	0.0139▲	0.0159▲	0.0166▲	0.00489▲
DAOA-QC0		⁄	0.111▲	0.117▲	0.119▲	0.120▲	0.108▲
DAOA-QC10			⁄	0.006▲	0.00795▲	0.00871▲	−0.00302 -
DAOA-QC25				⁄	0.00195 -	0.00271▲	−0.00902▽
DAOM-QC0					⁄	0.000758 -	−0.011▽
DAOM-QC10						⁄	−0.0117▽
DAOM-QC25							⁄

Table A.29: DAO-QC Convergence statistics for 1URR Cross comparison Average fitness

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−0.0173▽	0.00452▲	0.00765▲	0.00143 -	0.000327 -	−0.00352▽
DAOA-QC0		⁄	0.0218▲	0.0249▲	0.0187▲	0.0176▲	0.0138▲
DAOA-QC10			⁄	0.00313▲	−0.00309▽	−0.00419▽	−0.00804▽
DAOA-QC25				⁄	−0.00622▽	−0.00732▽	−0.0112▽
DAOM-QC0					⁄	−0.0011 -	−0.00495▽
DAOM-QC10						⁄	−0.00385▽
DAOM-QC25							⁄

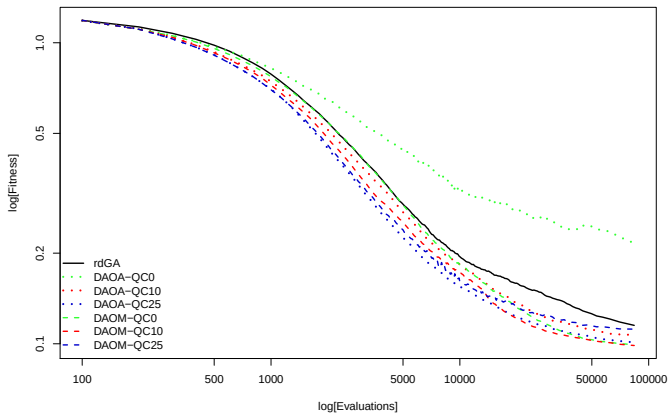
Table A.30: DAO-QC Convergence statistics for 1URR Cross comparison Best fitness

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−50.831▲	−36.680▲	−11.021▲	17.682▽	19.014▽	16.843▽
DAOA-QC0		⁄	14.151▽	39.810▽	68.512▽	69.845▽	67.674▽
DAOA-QC10			⁄	25.658▽	54.361▽	55.694▽	53.523▽
DAOA-QC25				⁄	28.703▽	30.035▽	27.865▽
DAOM-QC0					⁄	1.332▽	−0.838▲
DAOM-QC10						⁄	−2.170▲
DAOM-QC25							⁄

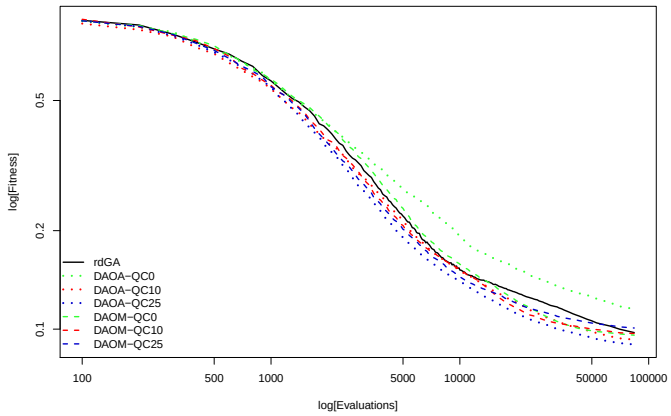
Table A.31: DAO-QC Convergence statistics for 1URR Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.115_{\pm 6.30e-03}$	$0.0971_{\pm 4.96e-03}$	$22.962_{\pm 2.32e+00}$
DAOA-QC0	$0.218_{\pm 2.49e-02}$	$0.114_{\pm 5.20e-03}$	$73.793_{\pm 1.82e+00}$
DAOA-QC10	$0.107_{\pm 5.31e-03}$	$0.0926_{\pm 2.99e-03}$	$59.642_{\pm 3.70e+00}$
DAOA-QC25	$0.101_{\pm 4.60e-03}$	$0.0895_{\pm 3.90e-03}$	$33.983_{\pm 5.13e+00}$
DAOM-QC0	$0.099_{\pm 4.41e-03}$	$0.0957_{\pm 4.00e-03}$	$5.280_{\pm 1.12e+00}$
DAOM-QC10	$0.0982_{\pm 5.77e-03}$	$0.0968_{\pm 5.61e-03}$	$3.948_{\pm 8.77e-01}$
DAOM-QC25	$0.110_{\pm 2.80e-02}$	$0.101_{\pm 6.45e-03}$	$6.118_{\pm 8.81e+00}$

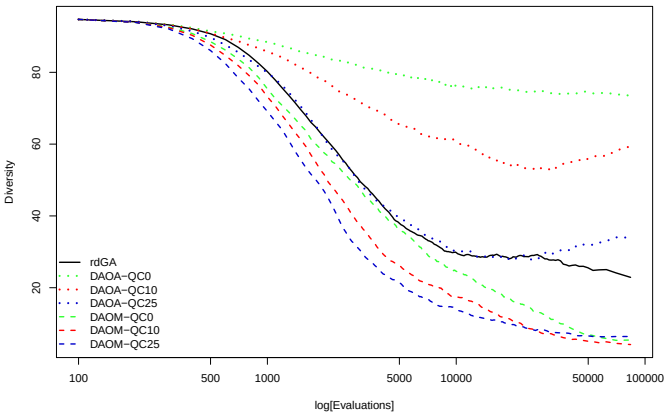
Table A.32: DAO-QC Convergence statistics for 1URR Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure A.8: Convergence analysis DAO-QC *1URR*

A.4 Preference Based Genetic Algorithm Study

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	⁄	−0.0239▽	0.0026 -	0.0208▲	0.0291▲	0.0341▲	0.00714▲
PBGA 0.3 0.7		⁄	0.0265▲	0.0446▲	0.053▲	0.0579▲	0.031▲
PBGA 0.5 0.5			⁄	0.0182▲	0.0265▲	0.0315▲	0.00454▲
PBGA 0.7 0.3				⁄	0.00832▲	0.0133▲	−0.0136▽
PBGA 0.8 0.2					⁄	0.00497▲	−0.0219▽
PBGA 0.9 0.1						⁄	−0.0269▽
PBGA 1.0 0.0							⁄

Table A.33: PBGA Convergence statistics for 1B3A Cross comparison Average fitness

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	⁄	−0.0293▽	−0.0105▽	0.00236 -	0.00803▲	0.0102▲	−0.0202▽
PBGA 0.3 0.7		⁄	0.0188▲	0.0317▲	0.0373▲	0.0395▲	0.00913▲
PBGA 0.5 0.5			⁄	0.0129▲	0.0185▲	0.0207▲	−0.00968 -
PBGA 0.7 0.3				⁄	0.00567▲	0.00784▲	−0.0225▽
PBGA 0.8 0.2					⁄	0.00216▲	−0.0282▽
PBGA 0.9 0.1						⁄	−0.0304▽
PBGA 1.0 0.0							⁄

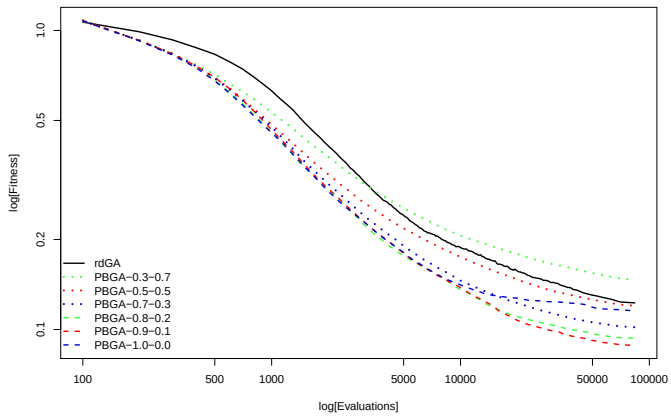
Table A.34: PBGA Convergence statistics for 1B3A Cross comparison Best fitness

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	⁄	−58.049▲	−54.347▲	−46.984▲	−39.074▲	−17.139▲	31.545▽
PBGA 0.3 0.7		⁄	3.702▽	11.065▽	18.975▽	40.910▽	89.594▽
PBGA 0.5 0.5			⁄	7.363▽	15.273▽	37.208▽	85.892▽
PBGA 0.7 0.3				⁄	7.909▽	29.844▽	78.529▽
PBGA 0.8 0.2					⁄	21.935▽	70.619▽
PBGA 0.9 0.1						⁄	48.684▽
PBGA 1.0 0.0							⁄

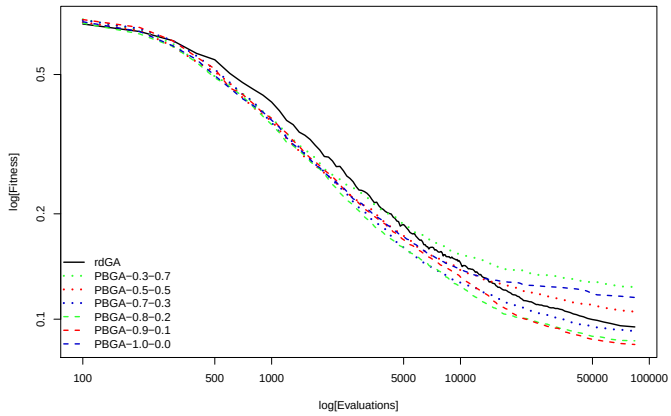
Table A.35: PBGA Convergence statistics for 1B3A Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.122_{\pm 1.14e-02}$	$0.0948_{\pm 9.42e-03}$	$31.886_{\pm 3.20e+00}$
PBGA 0.3 0.7	$0.146_{\pm 3.08e-03}$	$0.124_{\pm 3.50e-03}$	$89.936_{\pm 1.83e-01}$
PBGA 0.5 0.5	$0.120_{\pm 3.06e-03}$	$0.105_{\pm 3.50e-03}$	$86.234_{\pm 3.34e-01}$
PBGA 0.7 0.3	$0.102_{\pm 2.75e-03}$	$0.0924_{\pm 2.83e-03}$	$78.870_{\pm 7.28e-01}$
PBGA 0.8 0.2	$0.0932_{\pm 2.39e-03}$	$0.0867_{\pm 2.50e-03}$	$70.961_{\pm 1.08e+00}$
PBGA 0.9 0.1	$0.0883_{\pm 4.10e-03}$	$0.0846_{\pm 3.86e-03}$	$49.026_{\pm 2.80e+00}$
PBGA 1.0 0.0	$0.115_{\pm 2.05e-02}$	$0.115_{\pm 2.02e-02}$	$0.342_{\pm 6.47e-01}$

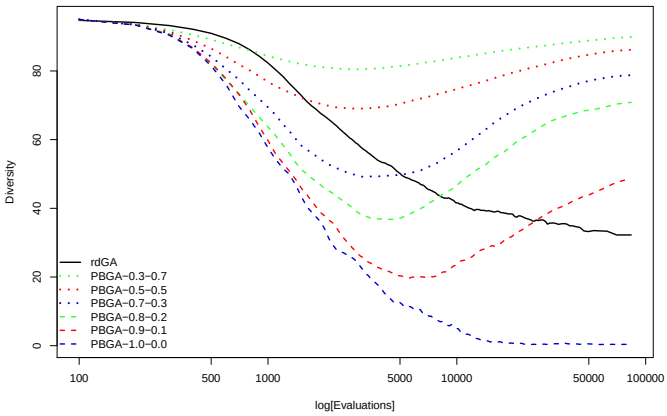
Table A.36: PBGA Convergence statistics for 1B3A Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure A.9: Convergence analysis PBGA *1B3A*

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	⁄	−0.0476▽	−0.0198▽	−0.00239▽	0.00594▲	0.012▲	−0.00321▽
PBGA 0.3 0.7		⁄	0.0279▲	0.0453▲	0.0536▲	0.0597▲	0.0444▲
PBGA 0.5 0.5			⁄	0.0174▲	0.0257▲	0.0318▲	0.0166▲
PBGA 0.7 0.3				⁄	0.00833▲	0.0144▲	−0.00083 -
PBGA 0.8 0.2					⁄	0.00607▲	−0.00916▽
PBGA 0.9 0.1						⁄	−0.0152▽
PBGA 1.0 0.0							⁄

Table A.37: PBGA Convergence statistics for 256B Cross comparison Average fitness

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	⁄	−0.0371▽	−0.0176▽	−0.00494▽	0.000943 -	0.00461▲	−0.014▽
PBGA 0.3 0.7		⁄	0.0195▲	0.0322▲	0.0381▲	0.0417▲	0.0231▲
PBGA 0.5 0.5			⁄	0.0127▲	0.0186▲	0.0222▲	0.00357▲
PBGA 0.7 0.3				⁄	0.00588▲	0.00954▲	−0.0091▽
PBGA 0.8 0.2					⁄	0.00366▲	−0.015▽
PBGA 0.9 0.1						⁄	−0.0186▽
PBGA 1.0 0.0							⁄

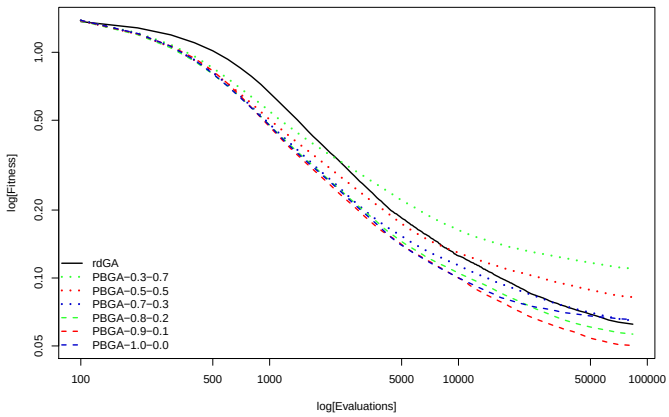
Table A.38: PBGA Convergence statistics for 256B Cross comparison Best fitness

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	⁄	−70.914▲	−67.481▲	−60.132▲	−52.360▲	−33.317▲	19.524▽
PBGA 0.3 0.7		⁄	3.433▽	10.783▽	18.555▽	37.597▽	90.438▽
PBGA 0.5 0.5			⁄	7.349▽	15.121▽	34.164▽	87.005▽
PBGA 0.7 0.3				⁄	7.772▽	26.815▽	79.656▽
PBGA 0.8 0.2					⁄	19.043▽	71.884▽
PBGA 0.9 0.1						⁄	52.841▽
PBGA 1.0 0.0							⁄

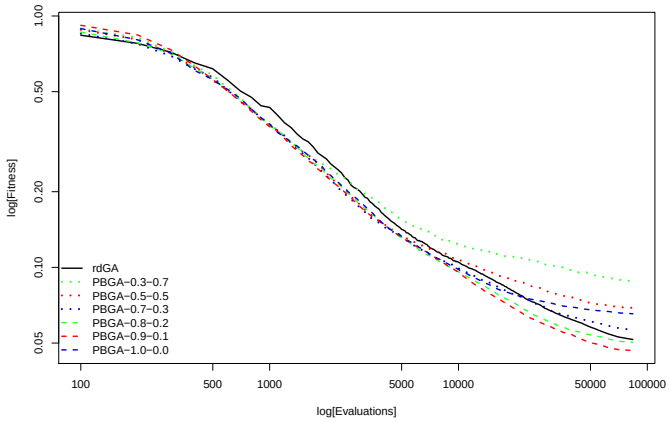
Table A.39: PBGA Convergence statistics for 256B Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.0621_{\pm 5.50e-03}$	$0.0512_{\pm 5.94e-03}$	$19.815_{\pm 2.26e+00}$
PBGA 0.3 0.7	$0.110_{\pm 2.09e-03}$	$0.0883_{\pm 3.44e-03}$	$90.730_{\pm 1.79e-01}$
PBGA 0.5 0.5	$0.0819_{\pm 1.71e-03}$	$0.0688_{\pm 2.12e-03}$	$87.296_{\pm 2.96e-01}$
PBGA 0.7 0.3	$0.0645_{\pm 1.92e-03}$	$0.0561_{\pm 1.91e-03}$	$79.947_{\pm 5.30e-01}$
PBGA 0.8 0.2	$0.0561_{\pm 2.63e-03}$	$0.0503_{\pm 2.52e-03}$	$72.175_{\pm 8.64e-01}$
PBGA 0.9 0.1	$0.0501_{\pm 3.86e-03}$	$0.0466_{\pm 3.87e-03}$	$53.132_{\pm 2.04e+00}$
PBGA 1.0 0.0	$0.0653_{\pm 6.82e-03}$	$0.0653_{\pm 6.82e-03}$	$0.291_{\pm 3.79e-01}$

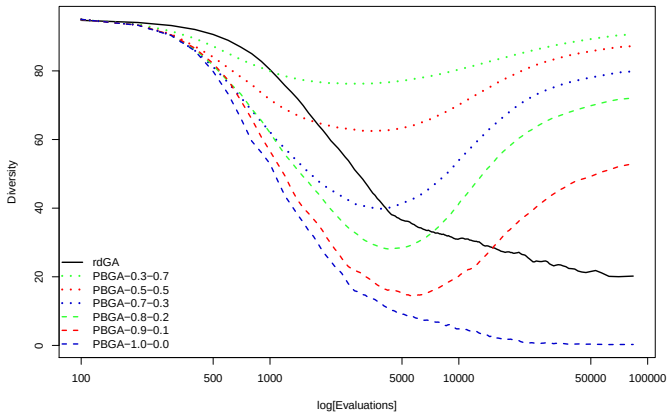
Table A.40: PBGA Convergence statistics for 256B Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure A.10: Convergence analysis PBGA *256B*

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	/	−0.0404▽	−0.016▽	−0.000333 -	0.00706▲	0.0149▲	0.0071▲
PBGA 0.3 0.7		/	0.0243▲	0.0401▲	0.0474▲	0.0552▲	0.0475▲
PBGA 0.5 0.5			/	0.0157▲	0.0231▲	0.0309▲	0.0231▲
PBGA 0.7 0.3				/	0.00739▲	0.0152▲	0.00743▲
PBGA 0.8 0.2					/	0.0078▲	$4.1e-05$ -
PBGA 0.9 0.1						/	−0.00775▽
PBGA 1.0 0.0							/

Table A.41: PBGA Convergence statistics for 1OAI Cross comparison Average fitness

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	/	−0.0368▽	−0.0193▽	−0.00814▽	−0.00237▽	0.00316▲	−0.00853▽
PBGA 0.3 0.7		/	0.0176▲	0.0287▲	0.0345▲	0.040▲	0.0283▲
PBGA 0.5 0.5			/	0.0111▲	0.0169▲	0.0224▲	0.0107▲
PBGA 0.7 0.3				/	0.00576▲	0.0113▲	−0.000395 -
PBGA 0.8 0.2					/	0.00554▲	−0.00616▽
PBGA 0.9 0.1						/	−0.0117▽
PBGA 1.0 0.0							/

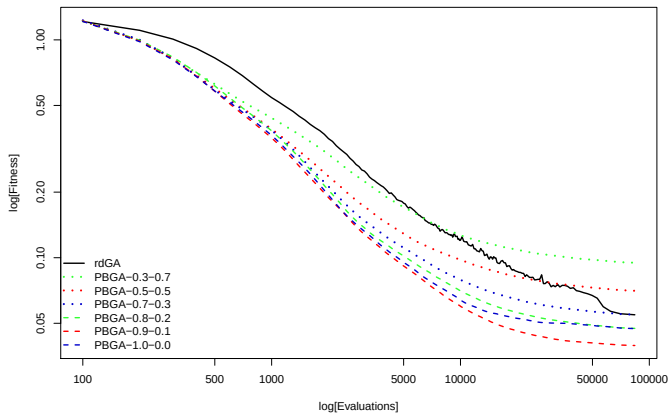
Table A.42: PBGA Convergence statistics for 1OAI Cross comparison Best fitness

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	/	−62.606▲	−59.075▲	−52.643▲	−45.782▲	−29.079▲	28.651▽
PBGA 0.3 0.7		/	3.530▽	9.963▽	16.824▽	33.527▽	91.257▽
PBGA 0.5 0.5			/	6.433▽	13.294▽	29.997▽	87.727▽
PBGA 0.7 0.3				/	6.861▽	23.564▽	81.294▽
PBGA 0.8 0.2					/	16.703▽	74.433▽
PBGA 0.9 0.1						/	57.730▽
PBGA 1.0 0.0							/

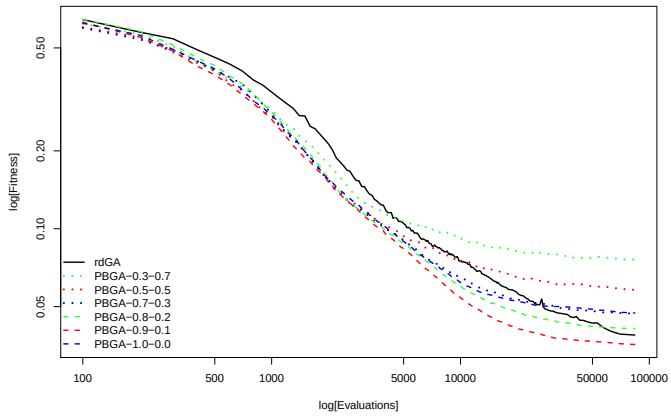
Table A.43: PBGA Convergence statistics for 1OAI Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.0543_{\pm 2.75e-03}$	$0.0387_{\pm 1.99e-03}$	$29.084_{\pm 2.10e+00}$
PBGA 0.3 0.7	$0.0947_{\pm 1.23e-03}$	$0.0755_{\pm 3.55e-03}$	$91.689_{\pm 1.11e-01}$
PBGA 0.5 0.5	$0.0704_{\pm 8.30e-04}$	$0.058_{\pm 2.14e-03}$	$88.159_{\pm 1.86e-01}$
PBGA 0.7 0.3	$0.0547_{\pm 1.15e-03}$	$0.0468_{\pm 1.94e-03}$	$81.727_{\pm 4.82e-01}$
PBGA 0.8 0.2	$0.0473_{\pm 1.32e-03}$	$0.0411_{\pm 1.63e-03}$	$74.866_{\pm 8.19e-01}$
PBGA 0.9 0.1	$0.0395_{\pm 1.43e-03}$	$0.0355_{\pm 1.59e-03}$	$58.162_{\pm 1.74e+00}$
PBGA 1.0 0.0	$0.0472_{\pm 7.15e-03}$	$0.0472_{\pm 7.15e-03}$	$0.432_{\pm 4.85e-01}$

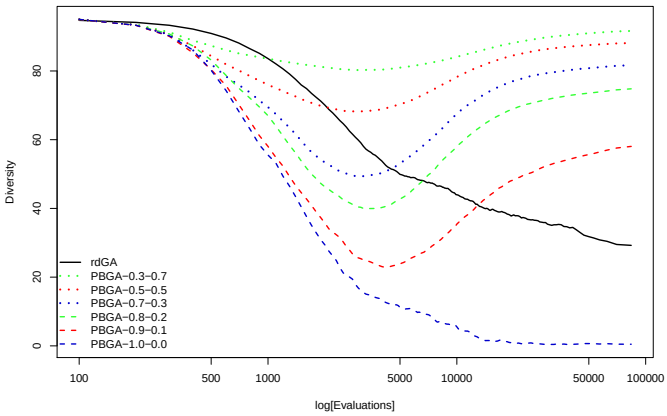
Table A.44: PBGA Convergence statistics for 1OAI Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure A.11: Convergence analysis PBGA *1OAI*

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	/	−0.0399▽	−0.0142▽	0.00558▲	0.0157▲	0.0224▲	0.00755▲
PBGA 0.3 0.7		/	0.0257▲	0.0455▲	0.0557▲	0.0624▲	0.0475▲
PBGA 0.5 0.5			/	0.0198▲	0.030▲	0.0366▲	0.0218▲
PBGA 0.7 0.3				/	0.0102▲	0.0169▲	0.00198▲
PBGA 0.8 0.2					/	0.00669▲	−0.00819▽
PBGA 0.9 0.1						/	−0.0149▽
PBGA 1.0 0.0							/

Table A.45: PBGA Convergence statistics for 1URR Cross comparison Average fitness

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	/	−0.0386▽	−0.0194▽	−0.00311▽	0.00424▲	0.00868▲	−0.0102▽
PBGA 0.3 0.7		/	0.0192▲	0.0355▲	0.0429▲	0.0473▲	0.0285▲
PBGA 0.5 0.5			/	0.0163▲	0.0237▲	0.0281▲	0.00928▲
PBGA 0.7 0.3				/	0.00735▲	0.0118▲	−0.00704▽
PBGA 0.8 0.2					/	0.00443▲	−0.0144▽
PBGA 0.9 0.1						/	−0.0188▽
PBGA 1.0 0.0							/

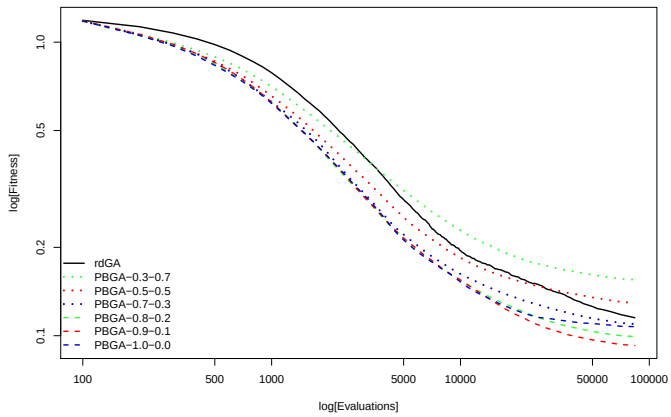
Table A.46: PBGA Convergence statistics for 1URR Cross comparison Best fitness

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	/	−66.224▲	−62.541▲	−54.706▲	−45.797▲	−23.482▲	22.846▽
PBGA 0.3 0.7		/	3.683▽	11.518▽	20.427▽	42.742▽	89.070▽
PBGA 0.5 0.5			/	7.835▽	16.745▽	39.060▽	85.388▽
PBGA 0.7 0.3				/	8.909▽	31.224▽	77.552▽
PBGA 0.8 0.2					/	22.315▽	68.643▽
PBGA 0.9 0.1						/	46.328▽
PBGA 1.0 0.0							/

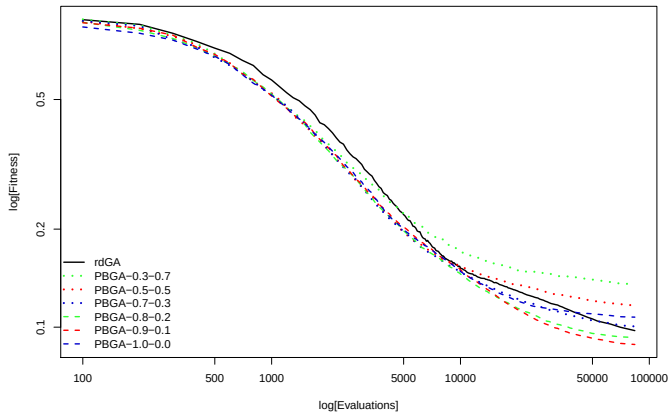
Table A.47: PBGA Convergence statistics for 1URR Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.115_{\pm 6.30e-03}$	$0.0971_{\pm 4.96e-03}$	$22.962_{\pm 2.32e+00}$
PBGA 0.3 0.7	$0.155_{\pm 1.85e-03}$	$0.136_{\pm 3.46e-03}$	$89.186_{\pm 1.77e-01}$
PBGA 0.5 0.5	$0.129_{\pm 2.05e-03}$	$0.117_{\pm 3.17e-03}$	$85.503_{\pm 2.42e-01}$
PBGA 0.7 0.3	$0.109_{\pm 2.21e-03}$	$0.100_{\pm 2.51e-03}$	$77.668_{\pm 5.33e-01}$
PBGA 0.8 0.2	$0.0991_{\pm 2.11e-03}$	$0.0929_{\pm 2.23e-03}$	$68.759_{\pm 8.30e-01}$
PBGA 0.9 0.1	$0.0924_{\pm 5.14e-03}$	$0.0885_{\pm 4.89e-03}$	$46.444_{\pm 3.18e+00}$
PBGA 1.0 0.0	$0.107_{\pm 9.52e-03}$	$0.107_{\pm 9.52e-03}$	$0.116_{\pm 2.04e-01}$

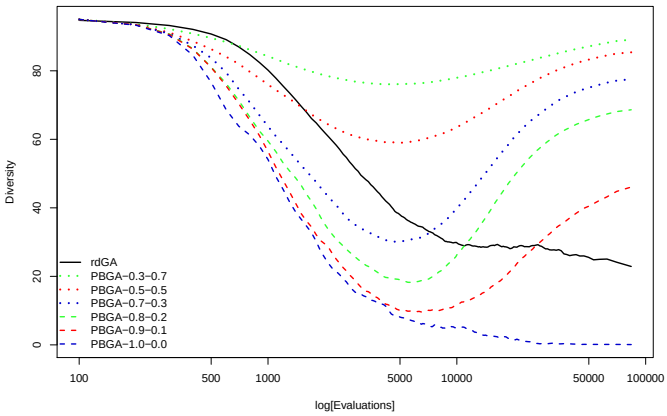
Table A.48: PBGA Convergence statistics for 1URR Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure A.12: Convergence analysis PBGA *1URR*

A.5 Summary of Algorithm Experiments Study

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	0.00599▲	0.0168▲	0.0204▲	0.0185▲	0.0291▲	0.0341▲
DAOA-QC10		⁄	0.0108▲	0.0144▲	0.0125▲	0.0231▲	0.0281▲
DAOA-QC25			⁄	0.00355▲	0.00165▲	0.0123▲	0.0172▲
DAOM-QC10				⁄	−0.00191 -	0.0087 -	0.0137▲
DAOM-QC25					⁄	0.0106▲	0.0156▲
PBGA 0.8 0.2						⁄	0.00497▲
PBGA 0.9 0.1							⁄

Table A.49: DAO vs. PBGA Convergence statistics for 1B3A Cross comparison Average fitness

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	0.00522▲	0.00856▲	−0.00495 -	−0.0047 -	0.00803▲	0.0102▲
DAOA-QC10		⁄	0.00333▲	−0.0102 -	−0.00993▽	0.00281▲	0.00498▲
DAOA-QC25			⁄	−0.0135▽	−0.0133▽	−0.000522 -	0.00164 -
DAOM-QC10				⁄	0.000249 -	0.013▲	0.0151▲
DAOM-QC25					⁄	0.0127▲	0.0149▲
PBGA 0.8 0.2						⁄	0.00216▲
PBGA 0.9 0.1							⁄

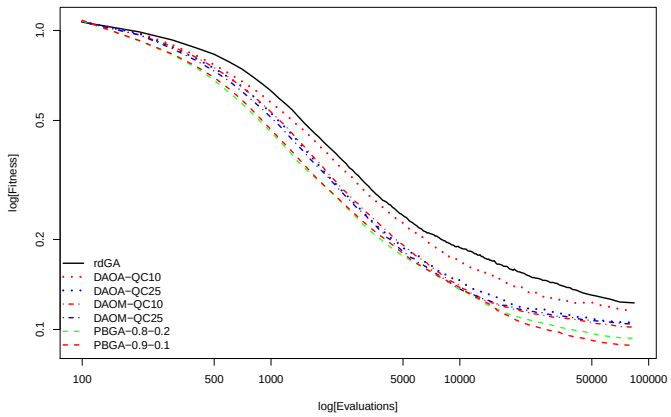
Table A.50: DAO vs. PBGA Convergence statistics for 1B3A Cross comparison Best fitness

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	−38.599▲	−18.534▲	25.954▽	25.355▽	−39.074▲	−17.139▲
DAOA-QC10		⁄	20.065▽	64.553▽	63.954▽	−0.476 -	21.459▽
DAOA-QC25			⁄	44.488▽	43.889▽	−20.541▲	1.394 -
DAOM-QC10				⁄	−0.599 -	−65.028▲	−43.093▲
DAOM-QC25					⁄	−64.429▲	−42.494▲
PBGA 0.8 0.2						⁄	21.935▽
PBGA 0.9 0.1							⁄

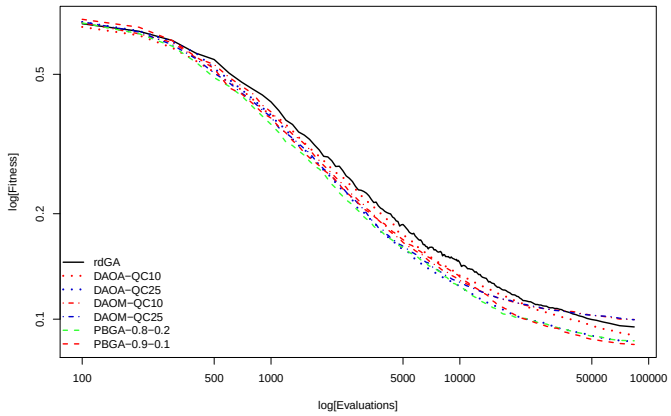
Table A.51: DAO vs. PBGA Convergence statistics for 1B3A Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.122_{\pm 1.14e-02}$	$0.0948_{\pm 9.42e-03}$	$31.886_{\pm 3.20e+00}$
DAOA-QC10	$0.116_{\pm 1.15e-02}$	$0.0895_{\pm 4.14e-03}$	$70.485_{\pm 4.63e+00}$
DAOA-QC25	$0.106_{\pm 7.35e-03}$	$0.0862_{\pm 4.36e-03}$	$50.420_{\pm 6.08e+00}$
DAOM-QC10	$0.102_{\pm 1.99e-02}$	$0.0997_{\pm 2.00e-02}$	$5.932_{\pm 1.30e+00}$
DAOM-QC25	$0.104_{\pm 1.71e-02}$	$0.0995_{\pm 1.72e-02}$	$6.531_{\pm 1.22e+00}$
PBGA 0.8 0.2	$0.0932_{\pm 2.39e-03}$	$0.0867_{\pm 2.50e-03}$	$70.961_{\pm 1.08e+00}$
PBGA 0.9 0.1	$0.0883_{\pm 4.10e-03}$	$0.0846_{\pm 3.86e-03}$	$49.026_{\pm 2.80e+00}$

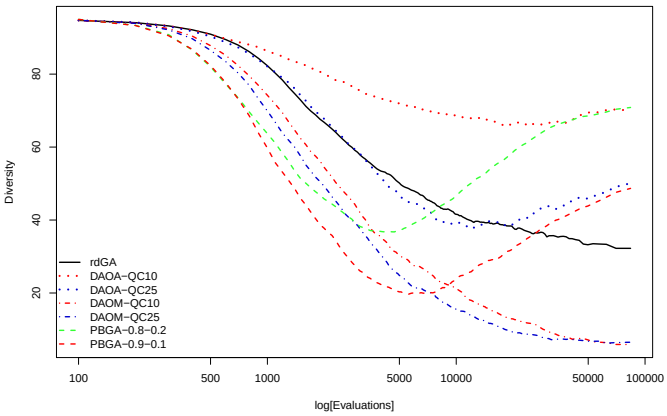
Table A.52: DAO vs. PBGA Convergence statistics for 1B3A Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure A.13: Convergence analysis DAO vs. PBGA *1B3A*

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	−0.00471▽	0.00642▲	0.00577▲	0.00503▲	0.00594▲	0.012▲
DAOA-QC10		⁄	0.0111▲	0.0105 -	0.00974 -	0.0107▲	0.0167▲
DAOA-QC25			⁄	−0.000646 -	−0.00139 -	−0.000474 -	0.0056▲
DAOM-QC10				⁄	−0.00074 -	0.000172 -	0.00625▲
DAOM-QC25					⁄	0.000912 -	0.00698▲
PBGA 0.8 0.2						⁄	0.00607▲
PBGA 0.9 0.1							⁄

Table A.53: DAO vs. PBGA Convergence statistics for 256B Cross comparison Average fitness

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	0.00292 -	0.00376▲	−0.00388▽	−0.00396▽	0.000943 -	0.00461▲
DAOA-QC10		⁄	0.000833 -	−0.00681▽	−0.00689▽	−0.00198▽	0.00168▲
DAOA-QC25			⁄	−0.00764▽	−0.00772▽	−0.00281▽	0.000849 -
DAOM-QC10				⁄	$-7.85e-05$ -	0.00483▲	0.00849▲
DAOM-QC25					⁄	0.00491▲	0.00857▲
PBGA 0.8 0.2						⁄	0.00366▲
PBGA 0.9 0.1							⁄

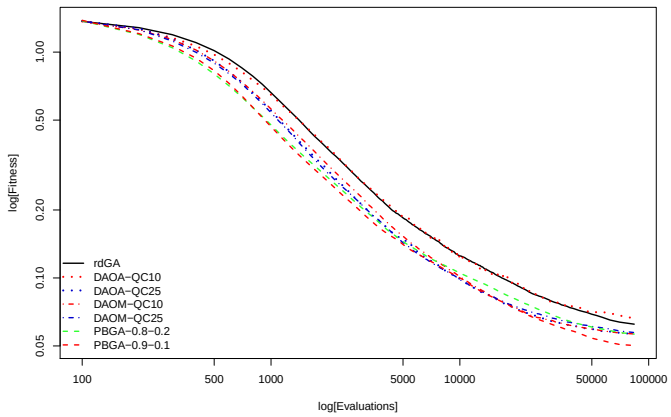
Table A.54: DAO vs. PBGA Convergence statistics for 256B Cross comparison Best fitness

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	−38.334▲	−16.675▲	15.387▽	15.485▽	−52.360▲	−33.317▲
DAOA-QC10		⁄	21.659▽	53.721▽	53.818▽	−14.026▲	5.017▽
DAOA-QC25			⁄	32.062▽	32.160▽	−35.685▲	−16.642▲
DAOM-QC10				⁄	0.0976 -	−67.747▲	−48.704▲
DAOM-QC25					⁄	−67.844▲	−48.802▲
PBGA 0.8 0.2						⁄	19.043▽
PBGA 0.9 0.1							⁄

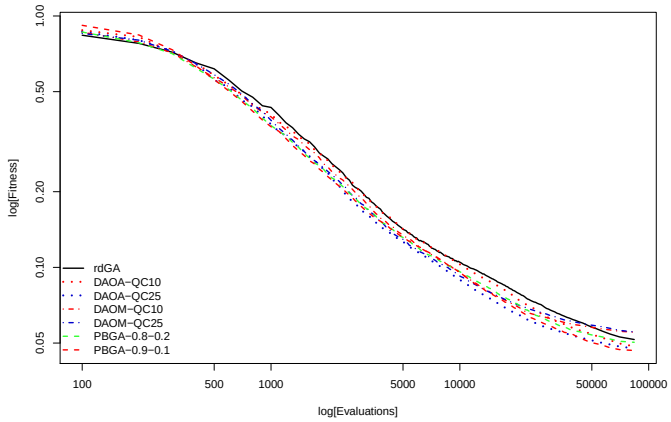
Table A.55: DAO vs. PBGA Convergence statistics for 256B Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.0621_{\pm 5.50e-03}$	$0.0512_{\pm 5.94e-03}$	$19.815_{\pm 2.26e+00}$
DAOA-QC10	$0.0668_{\pm 3.91e-02}$	$0.0483_{\pm 3.26e-03}$	$58.149_{\pm 6.47e+00}$
DAOA-QC25	$0.0557_{\pm 4.20e-03}$	$0.0475_{\pm 3.82e-03}$	$36.490_{\pm 6.29e+00}$
DAOM-QC10	$0.0563_{\pm 7.73e-03}$	$0.0551_{\pm 7.55e-03}$	$4.428_{\pm 9.44e-01}$
DAOM-QC25	$0.057_{\pm 5.64e-03}$	$0.0552_{\pm 5.42e-03}$	$4.331_{\pm 7.66e-01}$
PBGA 0.8 0.2	$0.0561_{\pm 2.63e-03}$	$0.0503_{\pm 2.52e-03}$	$72.175_{\pm 8.64e-01}$
PBGA 0.9 0.1	$0.0501_{\pm 3.86e-03}$	$0.0466_{\pm 3.87e-03}$	$53.132_{\pm 2.04e+00}$

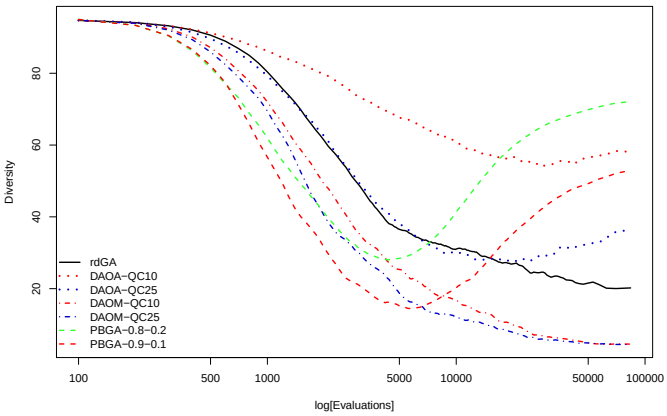
Table A.56: DAO vs. PBGA Convergence statistics for 256B Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure A.14: Convergence analysis DAO vs. PBGA *256B*

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	−0.00781▽	0.00187▲	0.0129▲	0.010▲	0.00706▲	0.0149▲
DAOA-QC10		⁄	0.00968▲	0.0207▲	0.0178▲	0.0149▲	0.0227▲
DAOA-QC25			⁄	0.011▲	0.00815▲	0.00519▲	0.013▲
DAOM-QC10				⁄	−0.00288▽	−0.00584▽	0.00196▲
DAOM-QC25					⁄	−0.00296▽	0.00483▲
PBGA 0.8 0.2						⁄	0.0078▲
PBGA 0.9 0.1							⁄

Table A.57: DAO vs. PBGA Convergence statistics for 1OAI Cross comparison Average fitness

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	−0.00113▽	0.00195▲	−0.00116 -	−0.00314▽	−0.00237▽	0.00316▲
DAOA-QC10		⁄	0.00308▲	$-2.81e-05$ -	−0.00201▽	−0.00124▽	0.00429▲
DAOA-QC25			⁄	−0.00311▽	−0.00508▽	−0.00432▽	0.00121▲
DAOM-QC10				⁄	−0.00198 -	−0.00121 -	0.00432▲
DAOM-QC25					⁄	0.000763 -	0.0063▲
PBGA 0.8 0.2						⁄	0.00554▲
PBGA 0.9 0.1							⁄

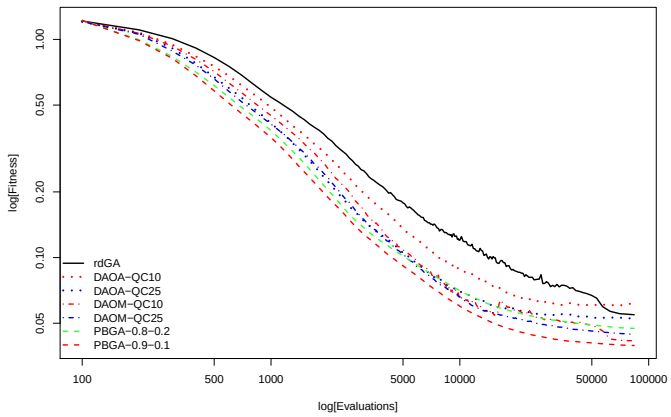
Table A.58: DAO vs. PBGA Convergence statistics for 1OAI Cross comparison Best fitness

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	−43.389▲	−22.895▲	21.455▽	21.549▽	−45.782▲	−29.079▲
DAOA-QC10		⁄	20.494▽	64.844▽	64.938▽	−2.393▲	14.310▽
DAOA-QC25			⁄	44.350▽	44.444▽	−22.887▲	−6.184▲
DAOM-QC10				⁄	0.094 -	−67.236▲	−50.533▲
DAOM-QC25					⁄	−67.330▲	−50.627▲
PBGA 0.8 0.2						⁄	16.703▽
PBGA 0.9 0.1							⁄

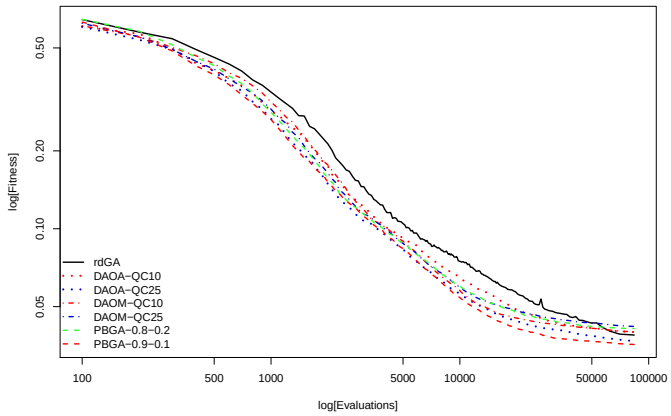
Table A.59: DAO vs. PBGA Convergence statistics for 1OAI Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.0543_{\pm 2.75e-03}$	$0.0387_{\pm 1.99e-03}$	$29.084_{\pm 2.10e+00}$
DAOA-QC10	$0.0621_{\pm 6.26e-03}$	$0.0398_{\pm 2.04e-03}$	$72.473_{\pm 3.08e+00}$
DAOA-QC25	$0.0525_{\pm 3.80e-03}$	$0.0368_{\pm 1.66e-03}$	$51.979_{\pm 5.24e+00}$
DAOM-QC10	$0.0414_{\pm 3.83e-03}$	$0.0399_{\pm 3.55e-03}$	$7.629_{\pm 1.92e+00}$
DAOM-QC25	$0.0443_{\pm 4.31e-03}$	$0.0418_{\pm 3.91e-03}$	$7.535_{\pm 7.91e-01}$
PBGA 0.8 0.2	$0.0473_{\pm 1.32e-03}$	$0.0411_{\pm 1.63e-03}$	$74.866_{\pm 8.19e-01}$
PBGA 0.9 0.1	$0.0395_{\pm 1.43e-03}$	$0.0355_{\pm 1.59e-03}$	$58.162_{\pm 1.74e+00}$

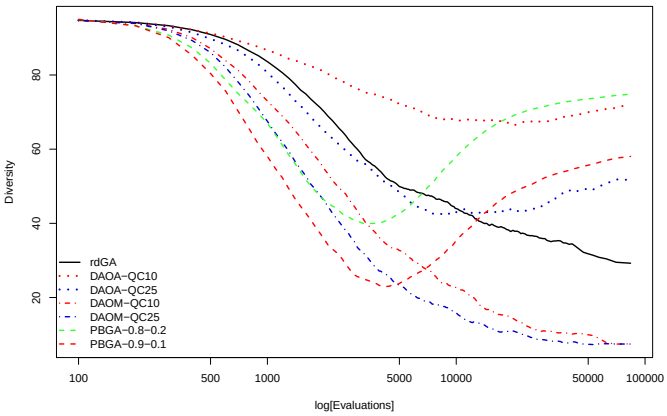
Table A.60: DAO vs. PBGA Convergence statistics for 1OAI Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure A.15: Convergence analysis DAO vs. PBGA *1OAI*

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	0.00791▲	0.0139▲	0.0166▲	0.00489▲	0.0157▲	0.0224▲
DAOA-QC10		⁄	0.006▲	0.00871▲	−0.00302 -	0.00783▲	0.0145▲
DAOA-QC25			⁄	0.00271▲	−0.00902▽	0.00183 -	0.00852▲
DAOM-QC10				⁄	−0.0117▽	−0.000882 -	0.00581▲
DAOM-QC25					⁄	0.0108▲	0.0175▲
PBGA 0.8 0.2						⁄	0.00669▲
PBGA 0.9 0.1							⁄

Table A.61: DAO vs. PBGA Convergence statistics for 1URR Cross comparison Average fitness

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	0.00452▲	0.00765▲	0.000327 -	−0.00352▽	0.00424▲	0.00868▲
DAOA-QC10		⁄	0.00313▲	−0.00419▽	−0.00804▽	−0.000275 -	0.00416▲
DAOA-QC25			⁄	−0.00732▽	−0.0112▽	−0.0034▽	0.00103 -
DAOM-QC10				⁄	−0.00385▽	0.00392▲	0.00835▲
DAOM-QC25					⁄	0.00777▲	0.0122▲
PBGA 0.8 0.2						⁄	0.00443▲
PBGA 0.9 0.1							⁄

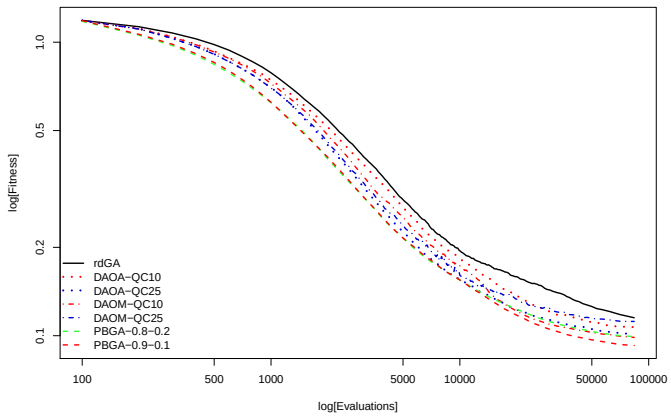
Table A.62: DAO vs. PBGA Convergence statistics for 1URR Cross comparison Best fitness

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	−36.680▲	−11.021▲	19.014▽	16.843▽	−45.797▲	−23.482▲
DAOA-QC10		⁄	25.658▽	55.694▽	53.523▽	−9.117▲	13.198▽
DAOA-QC25			⁄	30.035▽	27.865▽	−34.776▲	−12.461▲
DAOM-QC10				⁄	−2.170▲	−64.811▲	−42.496▲
DAOM-QC25					⁄	−62.640▲	−40.325▲
PBGA 0.8 0.2						⁄	22.315▽
PBGA 0.9 0.1							⁄

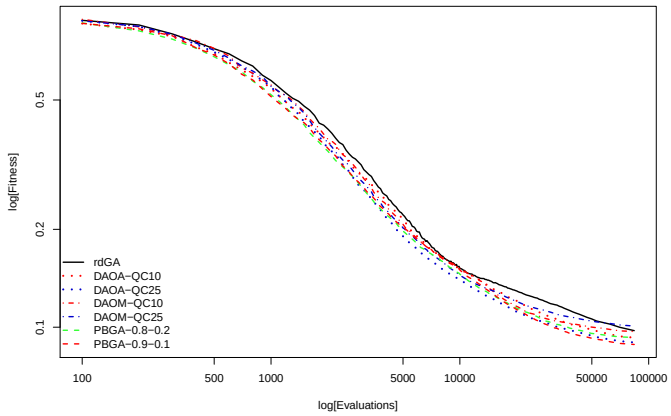
Table A.63: DAO vs. PBGA Convergence statistics for 1URR Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.115_{\pm 6.30e-03}$	$0.0971_{\pm 4.96e-03}$	$22.962_{\pm 2.32e+00}$
DAOA-QC10	$0.107_{\pm 5.31e-03}$	$0.0926_{\pm 2.99e-03}$	$59.642_{\pm 3.70e+00}$
DAOA-QC25	$0.101_{\pm 4.60e-03}$	$0.0895_{\pm 3.90e-03}$	$33.983_{\pm 5.13e+00}$
DAOM-QC10	$0.0982_{\pm 5.77e-03}$	$0.0968_{\pm 5.61e-03}$	$3.948_{\pm 8.77e-01}$
DAOM-QC25	$0.110_{\pm 2.80e-02}$	$0.101_{\pm 6.45e-03}$	$6.118_{\pm 8.81e+00}$
PBGA 0.8 0.2	$0.0991_{\pm 2.11e-03}$	$0.0929_{\pm 2.23e-03}$	$68.759_{\pm 8.30e-01}$
PBGA 0.9 0.1	$0.0924_{\pm 5.14e-03}$	$0.0885_{\pm 4.89e-03}$	$46.444_{\pm 3.18e+00}$

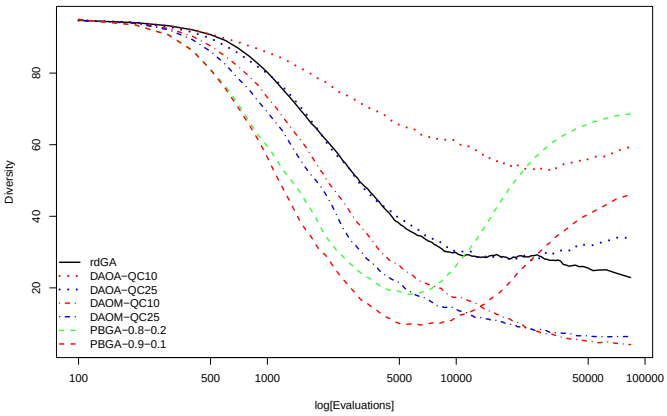
Table A.64: DAO vs. PBGA Convergence statistics for 1URR Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure A.16: Convergence analysis DAO vs. PBGA *1URR*

Appendix B

Detailed Results on Benchmarks

B.1 Description

The following four sections each contain one of four studies. The *Baseline Study* in Section B.2 compares selected existing genetic algorithms, the *Diversity as Objective with Quantile Constraint Study* in Section B.3 and *Preference Based Genetic Algorithm Study* in Section B.4 test the two developed algorithms with different settings. Section B.5 *Summary of Algorithm Experiments* summarises and compares both developed algorithms with a chosen baseline. Each study displays the same cross-comparison of final results and convergence plots for each sample in the study. The metrics studied are *Average Fitness*, *Best Fitness* and *Diversity* of 30 individual runs per algorithm and setting presented. For more details on the experiment setup, please refer to Chapter 9.

B.2 Baseline Study

	rdGA	gGA	scGA	ssGA
rdGA	⁄	0.0259▲	0.0919▲	0.0791▲
gGA		⁄	0.066▲	0.0532▲
scGA			⁄	−0.0128▽
ssGA				⁄

Table B.1: Base study convergence statistics for NK1-20-67 Cross comparison Average fitness

	rdGA	gGA	scGA	ssGA
rdGA	⁄	0.014▲	0.0173▲	0.00447 -
gGA		⁄	0.00332 -	−0.00949▽
scGA			⁄	−0.0128▽
ssGA				⁄

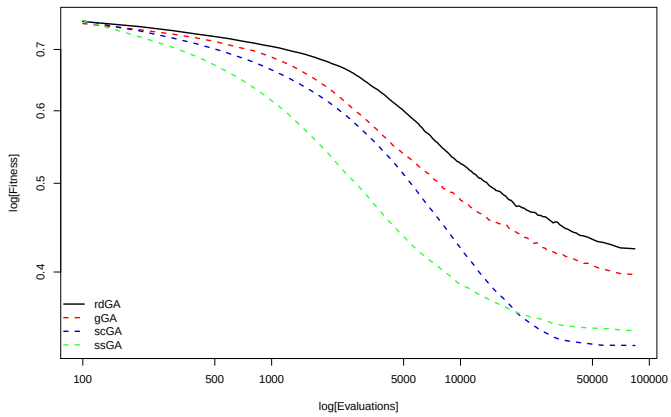
Table B.2: Base study convergence statistics for NK1-20-67 Cross comparison Best fitness

	rdGA	gGA	scGA	ssGA
rdGA	⁄	6.017▽	14.874▽	14.874▽
gGA		⁄	8.857▽	8.857▽
scGA			⁄	0.000⁄
ssGA				⁄

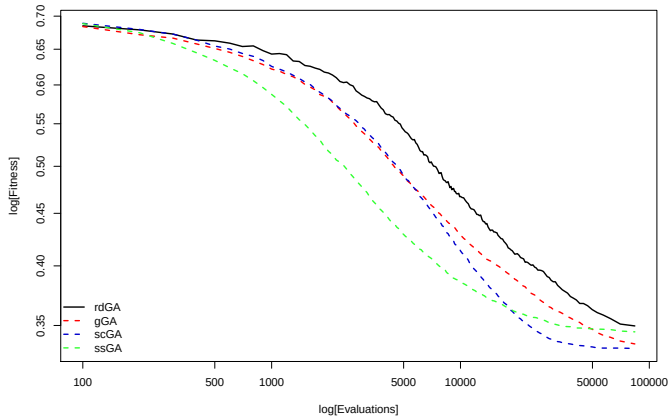
Table B.3: Base study convergence statistics for NK1-20-67 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.424_{\pm 1.61e-02}$	$0.350_{\pm 1.73e-02}$	$14.874_{\pm 1.81e+00}$
gGA	$0.398_{\pm 1.58e-02}$	$0.336_{\pm 1.49e-02}$	$8.857_{\pm 1.23e+00}$
scGA	$0.332_{\pm 1.59e-02}$	$0.332_{\pm 1.59e-02}$	$0.000_{\pm 0.00e+00}$
ssGA	$0.345_{\pm 1.78e-02}$	$0.345_{\pm 1.78e-02}$	$0.000_{\pm 0.00e+00}$

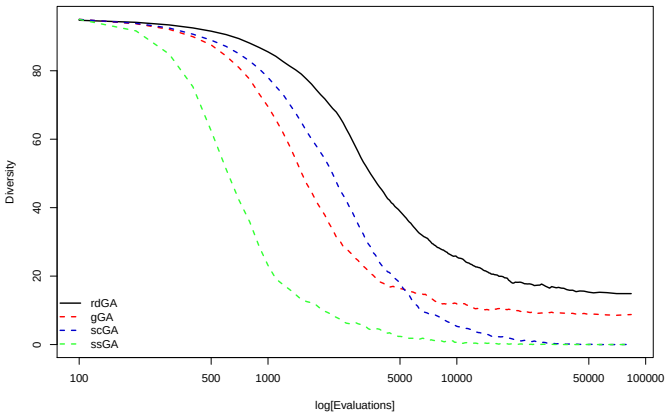
Table B.4: Base study convergence statistics for NK1-20-67 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.1: Convergence analysis base study *NK1-20-67*

	rdGA	gGA	scGA	ssGA
rdGA	⁄	0.0327▲	0.109▲	0.102▲
gGA		⁄	0.0765▲	0.0695▲
scGA			⁄	−0.00701 -
ssGA				⁄

Table B.5: Base study convergence statistics for NK2-20-67 Cross comparison Average fitness

	rdGA	gGA	scGA	ssGA
rdGA	⁄	0.0185▲	0.023▲	0.016▲
gGA		⁄	0.00457 -	−0.00244 -
scGA			⁄	−0.00701 -
ssGA				⁄

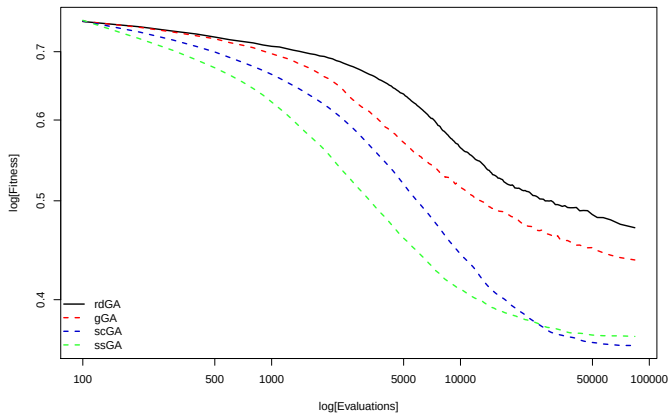
Table B.6: Base study convergence statistics for NK2-20-67 Cross comparison Best fitness

	rdGA	gGA	scGA	ssGA
rdGA	⁄	6.300▽	14.748▽	14.748▽
gGA		⁄	8.448▽	8.448▽
scGA			⁄	0.000⁄
ssGA				⁄

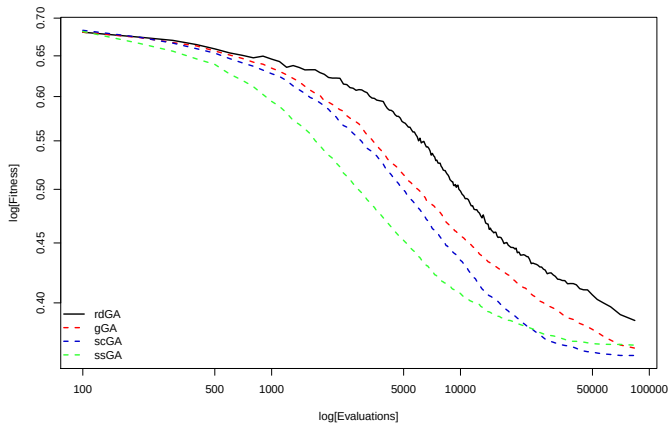
Table B.7: Base study convergence statistics for NK2-20-67 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.470_{\pm 1.37e-02}$	$0.384_{\pm 1.98e-02}$	$14.748_{\pm 1.63e+00}$
gGA	$0.437_{\pm 1.44e-02}$	$0.365_{\pm 1.92e-02}$	$8.448_{\pm 1.09e+00}$
scGA	$0.361_{\pm 1.78e-02}$	$0.361_{\pm 1.78e-02}$	$0.000_{\pm 0.00e+00}$
ssGA	$0.368_{\pm 1.83e-02}$	$0.368_{\pm 1.83e-02}$	$0.000_{\pm 0.00e+00}$

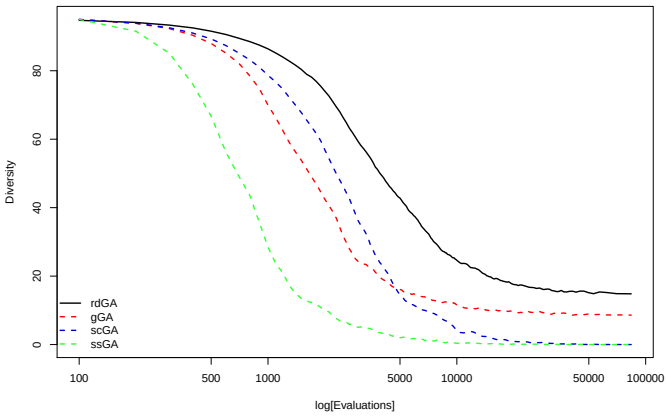
Table B.8: Base study convergence statistics for NK2-20-67 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.2: Convergence analysis base study *NK2-20-67*

	rdGA	gGA	scGA	ssGA
rdGA	⁄	−0.00802 -	0.0799▲	0.0375▲
gGA		⁄	0.0879▲	0.0456▲
scGA			⁄	−0.0424▽
ssGA				⁄

Table B.9: Base study convergence statistics for NK1-4-67 Cross comparison Average fitness

	rdGA	gGA	scGA	ssGA
rdGA	⁄	−0.0212▽	0.0139 -	−0.0285▽
gGA		⁄	0.0351▲	−0.00736 -
scGA			⁄	−0.0425▽
ssGA				⁄

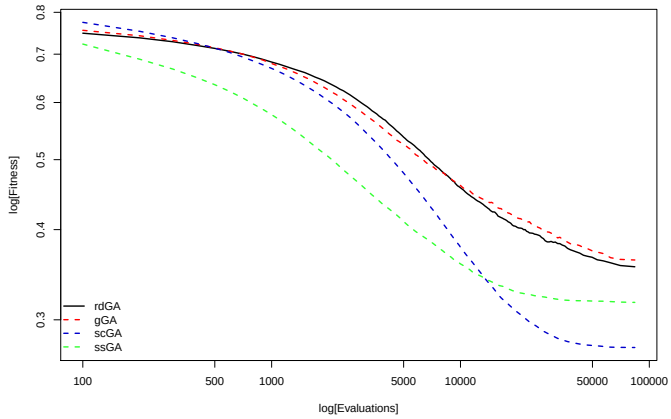
Table B.10: Base study convergence statistics for NK1-4-67 Cross comparison Best fitness

	rdGA	gGA	scGA	ssGA
rdGA	⁄	7.818▽	16.144▽	16.565▽
gGA		⁄	8.326▽	8.747▽
scGA			⁄	0.421▽
ssGA				⁄

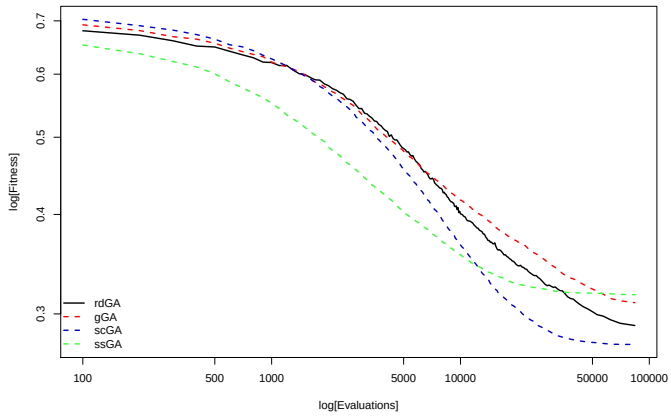
Table B.11: Base study convergence statistics for NK1-4-67 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.355_{\pm 3.82e-02}$	$0.288_{\pm 3.98e-02}$	$16.650_{\pm 1.49e+00}$
gGA	$0.363_{\pm 1.92e-02}$	$0.310_{\pm 2.17e-02}$	$8.831_{\pm 8.03e-01}$
scGA	$0.275_{\pm 1.90e-02}$	$0.275_{\pm 1.89e-02}$	$0.506_{\pm 5.50e-01}$
ssGA	$0.317_{\pm 1.78e-02}$	$0.317_{\pm 1.78e-02}$	$0.0846_{\pm 1.81e-01}$

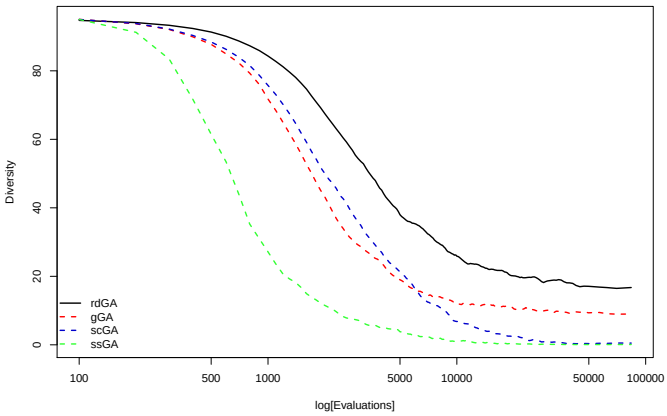
Table B.12: Base study convergence statistics for NK1-4-67 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.3: Convergence analysis base study *NK1-4-67*

	rdGA	gGA	scGA	ssGA
rdGA	⁄	0.0248▲	0.0778▲	0.0823▲
gGA		⁄	0.053▲	0.0575▲
scGA			⁄	0.0045 -
ssGA				⁄

Table B.13: Base study convergence statistics for NK1-20-97 Cross comparison Average fitness

	rdGA	gGA	scGA	ssGA
rdGA	⁄	0.0154▲	0.0263▲	0.0308▲
gGA		⁄	0.0109▲	0.0154▲
scGA			⁄	0.00454 -
ssGA				⁄

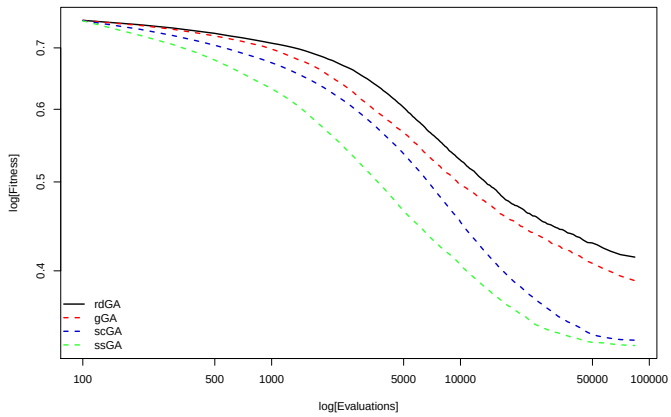
Table B.14: Base study convergence statistics for NK1-20-97 Cross comparison Best fitness

	rdGA	gGA	scGA	ssGA
rdGA	⁄	4.538▽	10.703▽	10.702▽
gGA		⁄	6.165▽	6.164▽
scGA			⁄	−0.000687 -
ssGA				⁄

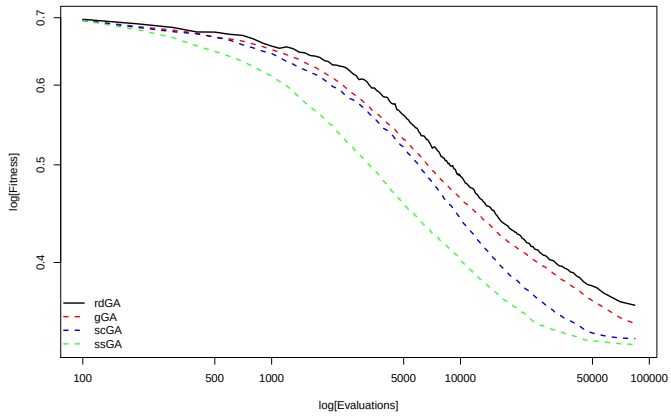
Table B.15: Base study convergence statistics for NK1-20-97 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.414_{\pm 1.10e-02}$	$0.362_{\pm 1.31e-02}$	$10.703_{\pm 1.28e+00}$
gGA	$0.389_{\pm 1.11e-02}$	$0.347_{\pm 1.24e-02}$	$6.165_{\pm 7.30e-01}$
scGA	$0.336_{\pm 1.61e-02}$	$0.336_{\pm 1.61e-02}$	$0.000_{\pm 0.00e+00}$
ssGA	$0.332_{\pm 1.61e-02}$	$0.332_{\pm 1.61e-02}$	$0.000687_{\pm 3.76e-03}$

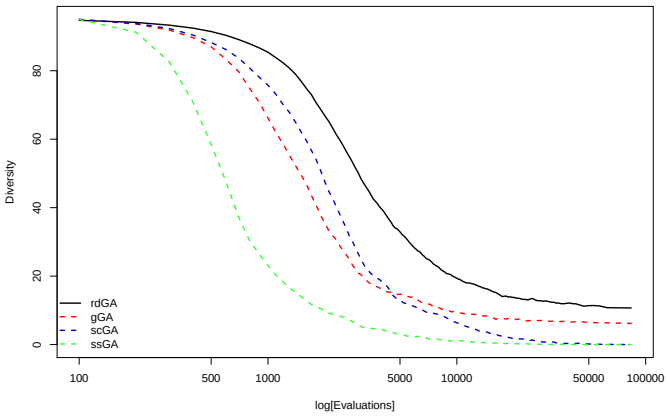
Table B.16: Base study convergence statistics for NK1-20-97 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.4: Convergence analysis base study *NK1-20-97*

	rdGA	gGA	scGA	ssGA
rdGA	⁄	0.0278▲	0.0843▲	0.0875▲
gGA		⁄	0.0565▲	0.0597▲
scGA			⁄	0.00319 -
ssGA				⁄

Table B.17: Base study convergence statistics for NK2-20-97 Cross comparison Average fitness

	rdGA	gGA	scGA	ssGA
rdGA	⁄	0.0169▲	0.0239▲	0.0271▲
gGA		⁄	0.00704 -	0.0102▲
scGA			⁄	0.00316 -
ssGA				⁄

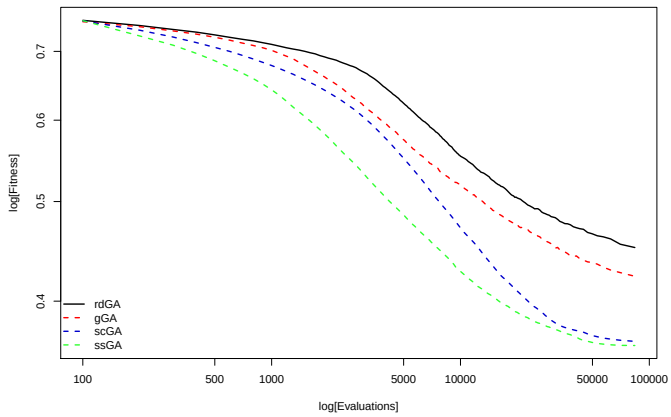
Table B.18: Base study convergence statistics for NK2-20-97 Cross comparison Best fitness

	rdGA	gGA	scGA	ssGA
rdGA	⁄	4.444▽	10.503▽	10.520▽
gGA		⁄	6.058▽	6.076▽
scGA			⁄	0.0173 -
ssGA				⁄

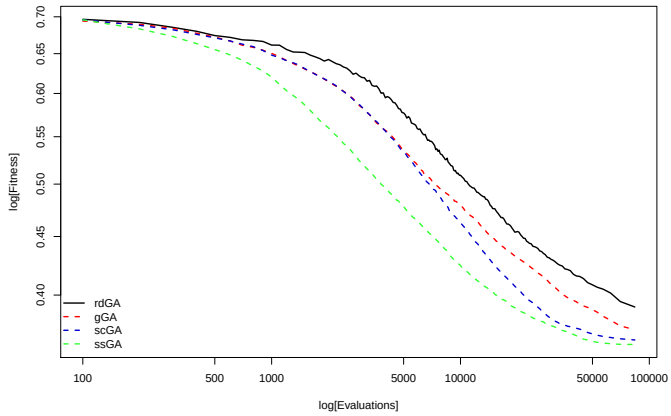
Table B.19: Base study convergence statistics for NK2-20-97 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.450_{\pm 1.03e-02}$	$0.389_{\pm 1.26e-02}$	$10.520_{\pm 1.04e+00}$
gGA	$0.422_{\pm 1.02e-02}$	$0.372_{\pm 9.93e-03}$	$6.076_{\pm 6.88e-01}$
scGA	$0.365_{\pm 1.31e-02}$	$0.365_{\pm 1.31e-02}$	$0.0173_{\pm 9.47e-02}$
ssGA	$0.362_{\pm 1.45e-02}$	$0.362_{\pm 1.45e-02}$	$0.000_{\pm 0.00e+00}$

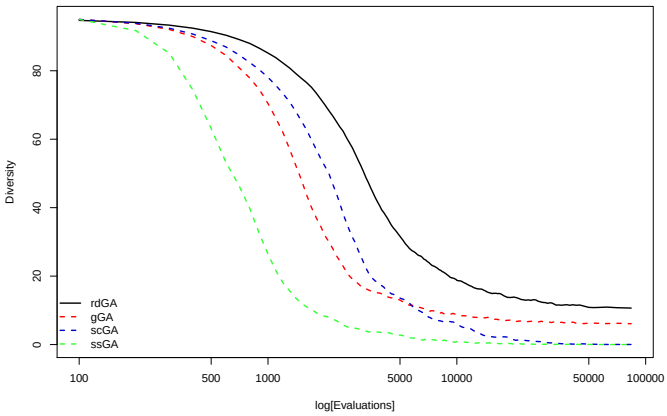
Table B.20: Base study convergence statistics for NK2-20-97 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.5: Convergence analysis base study *NK2-20-97*

	rdGA	gGA	scGA	ssGA
rdGA	⁄	−0.00958 -	0.0171▲	0.0624▲
gGA		⁄	0.0266▲	0.072▲
scGA			⁄	0.0454▲
ssGA				⁄

Table B.21: Base study convergence statistics for NK1-4-97 Cross comparison Average fitness

	rdGA	gGA	scGA	ssGA
rdGA	⁄	−0.0197▽	−0.0283▽	0.017▲
gGA		⁄	−0.00865▽	0.0367▲
scGA			⁄	0.0453▲
ssGA				⁄

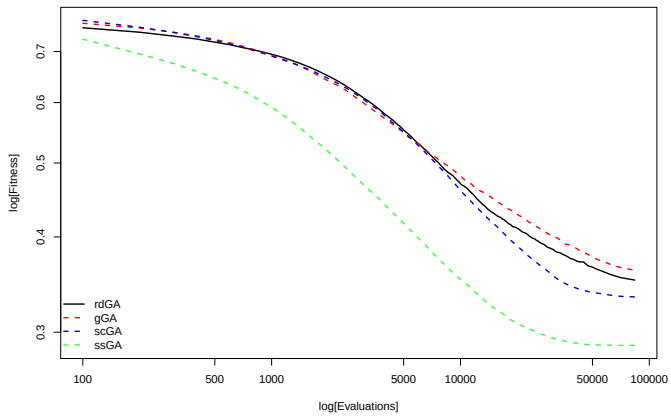
Table B.22: Base study convergence statistics for NK1-4-97 Cross comparison Best fitness

	rdGA	gGA	scGA	ssGA
rdGA	⁄	6.051▽	12.028▽	12.355▽
gGA		⁄	5.977▽	6.304▽
scGA			⁄	0.327▽
ssGA				⁄

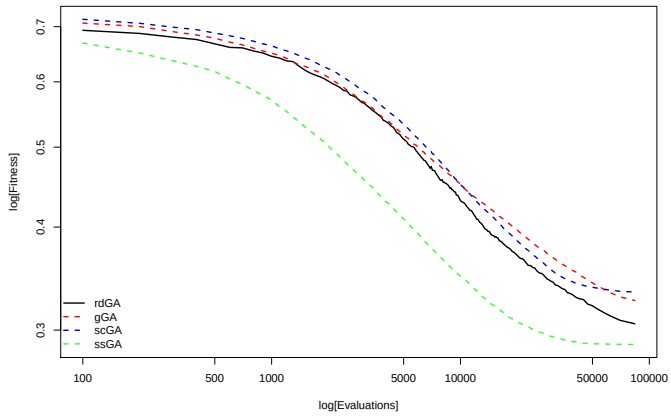
Table B.23: Base study convergence statistics for NK1-4-97 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.351_{\pm 2.50e-02}$	$0.305_{\pm 2.53e-02}$	$12.376_{\pm 1.06e+00}$
gGA	$0.360_{\pm 1.40e-02}$	$0.325_{\pm 1.31e-02}$	$6.325_{\pm 7.10e-01}$
scGA	$0.333_{\pm 1.58e-02}$	$0.333_{\pm 1.58e-02}$	$0.348_{\pm 3.26e-01}$
ssGA	$0.288_{\pm 1.57e-02}$	$0.288_{\pm 1.57e-02}$	$0.0213_{\pm 8.11e-02}$

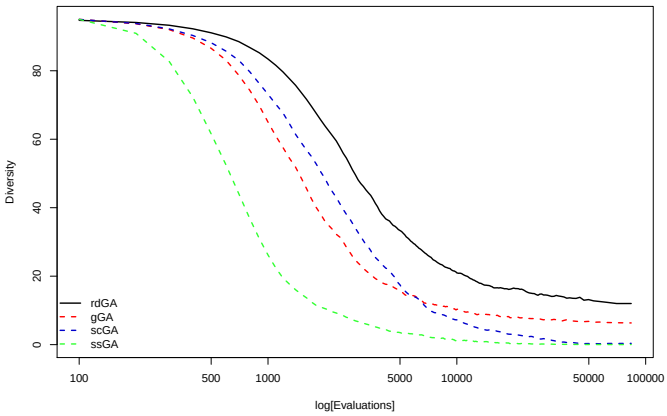
Table B.24: Base study convergence statistics for NK1-4-97 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.6: Convergence analysis base study *NK1-4-97*

	rdGA	gGA	scGA	ssGA
rdGA	⁄	0.0286▲	0.0624▲	0.0697▲
gGA		⁄	0.0339▲	0.0411▲
scGA			⁄	0.00723▲
ssGA				⁄

Table B.25: Base study convergence statistics for NK2-4-97 Cross comparison Average fitness

	rdGA	gGA	scGA	ssGA
rdGA	⁄	0.020▲	0.0105▲	0.0177▲
gGA		⁄	−0.00952▽	−0.0023 -
scGA			⁄	0.00722▲
ssGA				⁄

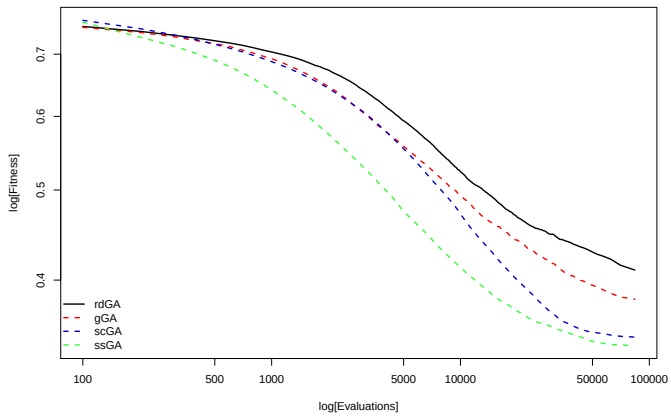
Table B.26: Base study convergence statistics for NK2-4-97 Cross comparison Best fitness

	rdGA	gGA	scGA	ssGA
rdGA	⁄	4.953▽	11.330▽	11.401▽
gGA		⁄	6.377▽	6.447▽
scGA			⁄	0.0707 -
ssGA				⁄

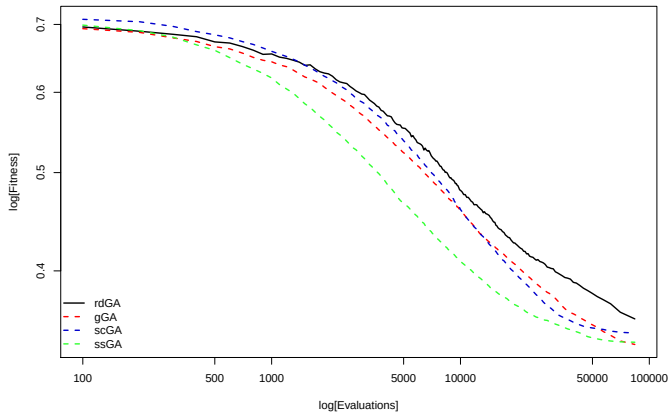
Table B.27: Base study convergence statistics for NK2-4-97 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.410_{\pm 1.61e-02}$	$0.358_{\pm 1.79e-02}$	$11.422_{\pm 8.10e-01}$
gGA	$0.381_{\pm 1.40e-02}$	$0.338_{\pm 1.42e-02}$	$6.468_{\pm 6.75e-01}$
scGA	$0.347_{\pm 1.51e-02}$	$0.347_{\pm 1.51e-02}$	$0.0918_{\pm 2.15e-01}$
ssGA	$0.340_{\pm 1.69e-02}$	$0.340_{\pm 1.69e-02}$	$0.0211_{\pm 6.71e-02}$

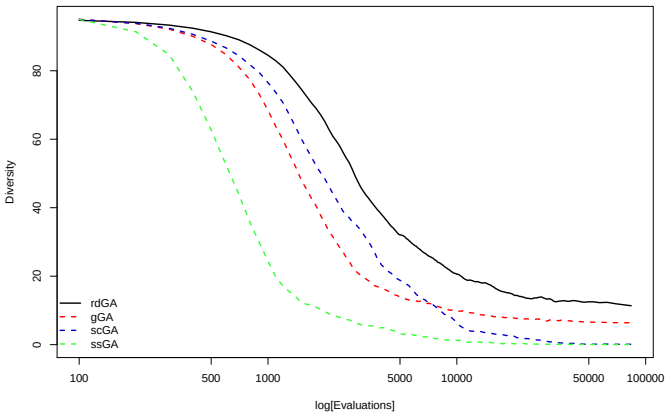
Table B.28: Base study convergence statistics for NK2-4-97 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.7: Convergence analysis base study *NK2-4-97*

B.3 Diversity as Objective with Quantile Constraint Study

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−0.114▽	−0.00988▽	0.0287▲	0.0673▲	0.076▲	0.0697▲
DAOA-QC0		⁄	0.104▲	0.142▲	0.181▲	0.190▲	0.184▲
DAOA-QC10			⁄	0.0386▲	0.0772▲	0.0859▲	0.0796▲
DAOA-QC25				⁄	0.0387▲	0.0473▲	0.041▲
DAOM-QC0					⁄	0.00866▲	0.00237 -
DAOM-QC10						⁄	−0.00629▽
DAOM-QC25							⁄

Table B.29: DAO-QC Convergence statistics for NK1-20-67 Cross comparison Average fitness

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−0.0599▽	−0.0198▽	0.000614 -	0.00668 -	0.0062 -	0.00703 -
DAOA-QC0		⁄	0.0401▲	0.0605▲	0.0666▲	0.0661▲	0.0669▲
DAOA-QC10			⁄	0.0204▲	0.0265▲	0.026▲	0.0268▲
DAOA-QC25				⁄	0.00607 -	0.00559 -	0.00642 -
DAOM-QC0					⁄	−0.000481 -	0.000347 -
DAOM-QC10						⁄	0.000828 -
DAOM-QC25							⁄

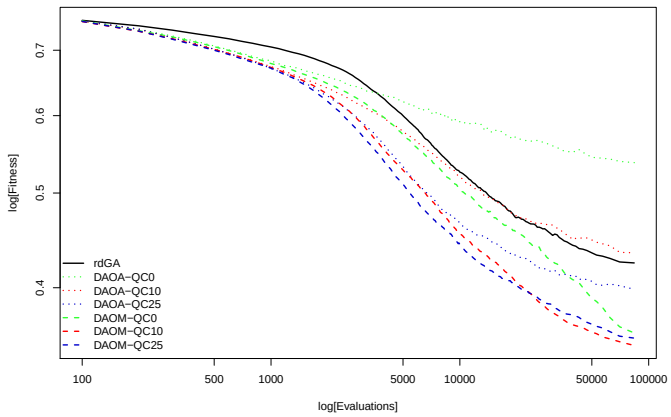
Table B.30: DAO-QC Convergence statistics for NK1-20-67 Cross comparison Best fitness

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−60.435▲	−37.734▲	−16.130▲	8.062▽	10.031▽	9.291▽
DAOA-QC0		⁄	22.701▽	44.305▽	68.497▽	70.466▽	69.726▽
DAOA-QC10			⁄	21.604▽	45.796▽	47.765▽	47.025▽
DAOA-QC25				⁄	24.193▽	26.161▽	25.421▽
DAOM-QC0					⁄	1.968▽	1.228▽
DAOM-QC10						⁄	−0.740▲
DAOM-QC25							⁄

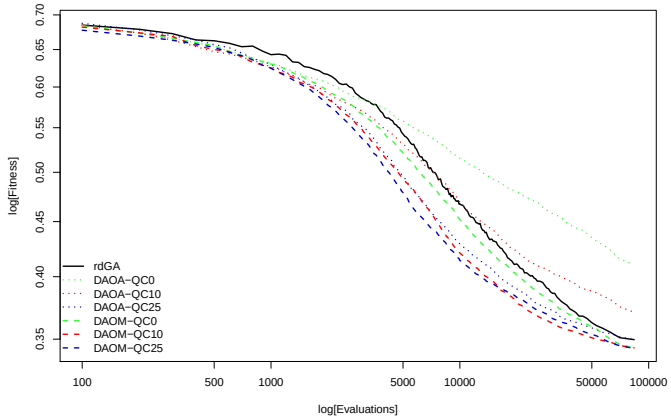
Table B.31: DAO-QC Convergence statistics for NK1-20-67 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.424_{\pm 1.61e-02}$	$0.350_{\pm 1.73e-02}$	$14.874_{\pm 1.81e+00}$
DAOA-QC0	$0.538_{\pm 1.14e-02}$	$0.409_{\pm 1.51e-02}$	$75.309_{\pm 1.75e+00}$
DAOA-QC10	$0.434_{\pm 1.73e-02}$	$0.369_{\pm 1.79e-02}$	$52.608_{\pm 7.00e+00}$
DAOA-QC25	$0.395_{\pm 1.60e-02}$	$0.349_{\pm 1.55e-02}$	$31.004_{\pm 5.16e+00}$
DAOM-QC0	$0.357_{\pm 1.29e-02}$	$0.343_{\pm 1.30e-02}$	$6.811_{\pm 1.48e+00}$
DAOM-QC10	$0.348_{\pm 1.67e-02}$	$0.343_{\pm 1.68e-02}$	$4.843_{\pm 9.24e-01}$
DAOM-QC25	$0.354_{\pm 1.70e-02}$	$0.342_{\pm 1.69e-02}$	$5.583_{\pm 1.05e+00}$

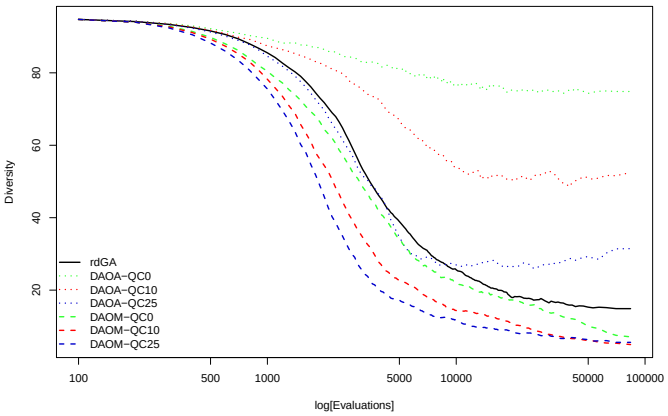
Table B.32: DAO-QC Convergence statistics for NK1-20-67 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.8: Convergence analysis DAO-QC *NK1-20-67*

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−0.0885▽	0.00709 -	0.0337▲	0.0834▲	0.0948▲	0.0831▲
DAOA-QC0		⁄	0.0955▲	0.122▲	0.172▲	0.183▲	0.172▲
DAOA-QC10			⁄	0.0266▲	0.0763▲	0.0877▲	0.076▲
DAOA-QC25				⁄	0.0497▲	0.061▲	0.0494▲
DAOM-QC0					⁄	0.0114▲	−0.000291 -
DAOM-QC10						⁄	−0.0117▽
DAOM-QC25							⁄

Table B.33: DAO-QC Convergence statistics for NK2-20-67 Cross comparison Average fitness

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−0.0606▽	−0.0175▽	0.00665 -	0.00874▲	0.0159▲	0.0098▲
DAOA-QC0		⁄	0.0431▲	0.0673▲	0.0694▲	0.0766▲	0.0704▲
DAOA-QC10			⁄	0.0242▲	0.0263▲	0.0335▲	0.0273▲
DAOA-QC25				⁄	0.00209 -	0.0093▲	0.00316 -
DAOM-QC0					⁄	0.0072 -	0.00106 -
DAOM-QC10						⁄	−0.00614 -
DAOM-QC25							⁄

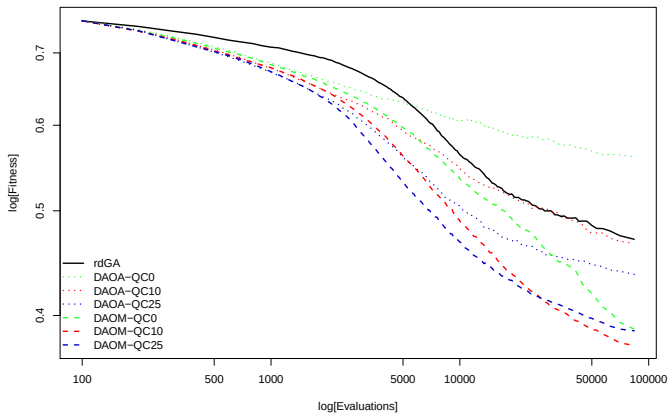
Table B.34: DAO-QC Convergence statistics for NK2-20-67 Cross comparison Best fitness

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−60.808▲	−37.853▲	−13.844▲	8.838▽	9.602▽	9.319▽
DAOA-QC0		⁄	22.956▽	46.964▽	69.647▽	70.410▽	70.127▽
DAOA-QC10			⁄	24.008▽	46.691▽	47.454▽	47.172▽
DAOA-QC25				⁄	22.683▽	23.446▽	23.163▽
DAOM-QC0					⁄	0.763▽	0.481 -
DAOM-QC10						⁄	−0.283▲
DAOM-QC25							⁄

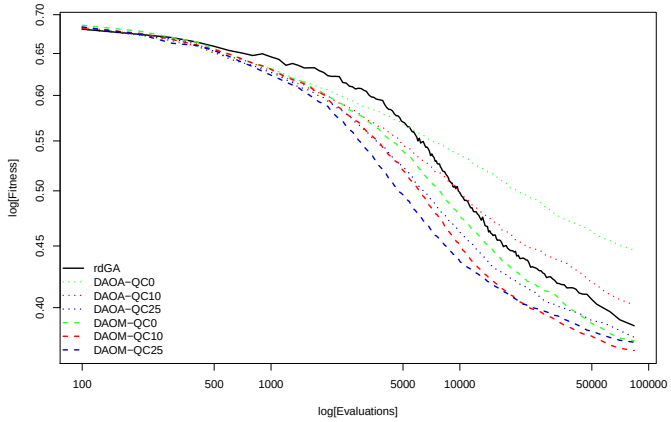
Table B.35: DAO-QC Convergence statistics for NK2-20-67 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.470_{\pm 1.37e-02}$	$0.384_{\pm 1.98e-02}$	$14.748_{\pm 1.63e+00}$
DAOA-QC0	$0.558_{\pm 1.52e-02}$	$0.444_{\pm 1.37e-02}$	$75.556_{\pm 2.84e+00}$
DAOA-QC10	$0.463_{\pm 2.38e-02}$	$0.401_{\pm 1.75e-02}$	$52.601_{\pm 1.01e+01}$
DAOA-QC25	$0.436_{\pm 1.45e-02}$	$0.377_{\pm 1.66e-02}$	$28.592_{\pm 4.62e+00}$
DAOM-QC0	$0.386_{\pm 1.57e-02}$	$0.375_{\pm 1.50e-02}$	$5.910_{\pm 1.08e+00}$
DAOM-QC10	$0.375_{\pm 1.57e-02}$	$0.368_{\pm 1.39e-02}$	$5.146_{\pm 1.36e+00}$
DAOM-QC25	$0.387_{\pm 1.51e-02}$	$0.374_{\pm 1.55e-02}$	$5.429_{\pm 8.93e-01}$

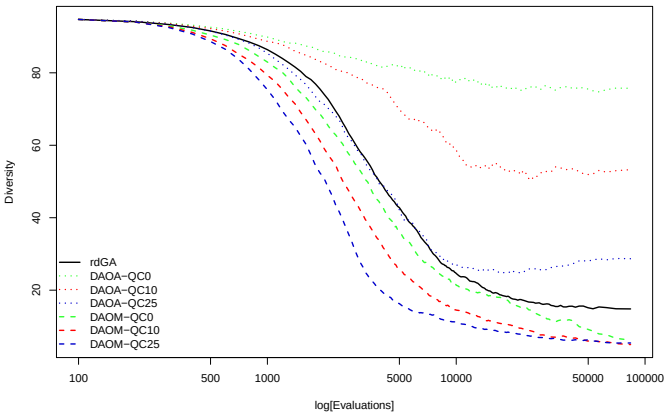
Table B.36: DAO-QC Convergence statistics for NK2-20-67 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.9: Convergence analysis DAO-QC *NK2-20-67*

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−0.138▽	−0.00521 -	0.0283▲	0.0385▲	0.0604▲	0.0512▲
DAOA-QC0		⁄	0.133▲	0.167▲	0.177▲	0.199▲	0.190▲
DAOA-QC10			⁄	0.0335▲	0.0437▲	0.0657▲	0.0564▲
DAOA-QC25				⁄	0.0102▲	0.0321▲	0.0229▲
DAOM-QC0					⁄	0.0219▲	0.0127▲
DAOM-QC10						⁄	−0.00921▽
DAOM-QC25							⁄

Table B.37: DAO-QC Convergence statistics for NK1-20-97 Cross comparison Average fitness

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−0.0762▽	−0.0126▽	0.014▲	0.00403 -	0.0147▲	0.00885▲
DAOA-QC0		⁄	0.0636▲	0.0902▲	0.0802▲	0.0909▲	0.085▲
DAOA-QC10			⁄	0.0266▲	0.0166▲	0.0273▲	0.0215▲
DAOA-QC25				⁄	−0.00995▽	0.000699 -	−0.00514 -
DAOM-QC0					⁄	0.0107▲	0.00481 -
DAOM-QC10						⁄	−0.00584 -
DAOM-QC25							⁄

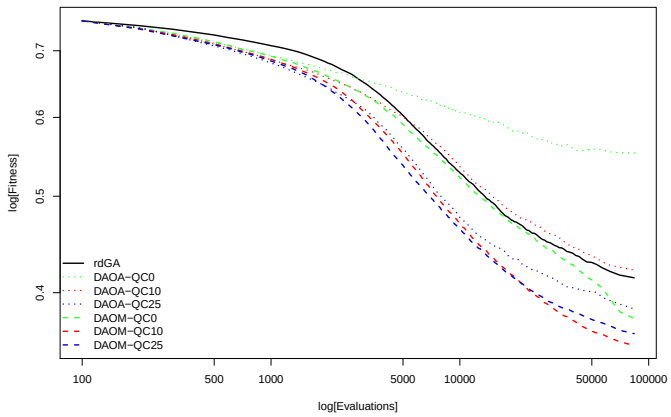
Table B.38: DAO-QC Convergence statistics for NK1-20-97 Cross comparison Best fitness

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−61.156▲	−29.333▲	−9.563▲	4.964▽	6.659▽	6.567▽
DAOA-QC0		⁄	31.824▽	51.593▽	66.120▽	67.816▽	67.724▽
DAOA-QC10			⁄	19.770▽	34.297▽	35.992▽	35.900▽
DAOA-QC25				⁄	14.527▽	16.222▽	16.130▽
DAOM-QC0					⁄	1.696▽	1.603▽
DAOM-QC10						⁄	−0.0922 -
DAOM-QC25							⁄

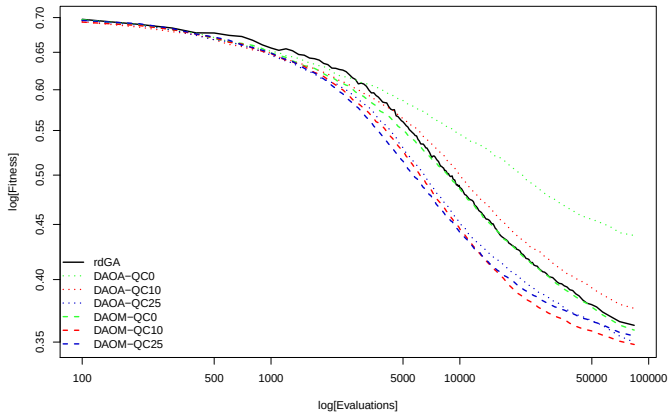
Table B.39: DAO-QC Convergence statistics for NK1-20-97 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.414_{\pm 1.10e-02}$	$0.362_{\pm 1.31e-02}$	$10.703_{\pm 1.28e+00}$
DAOA-QC0	$0.552_{\pm 1.48e-02}$	$0.439_{\pm 1.49e-02}$	$71.859_{\pm 2.30e+00}$
DAOA-QC10	$0.419_{\pm 1.49e-02}$	$0.375_{\pm 1.38e-02}$	$40.035_{\pm 6.33e+00}$
DAOA-QC25	$0.386_{\pm 1.54e-02}$	$0.348_{\pm 1.51e-02}$	$20.266_{\pm 3.34e+00}$
DAOM-QC0	$0.375_{\pm 1.60e-02}$	$0.358_{\pm 1.35e-02}$	$5.739_{\pm 2.00e+00}$
DAOM-QC10	$0.353_{\pm 1.31e-02}$	$0.348_{\pm 1.34e-02}$	$4.043_{\pm 9.29e-01}$
DAOM-QC25	$0.363_{\pm 1.17e-02}$	$0.354_{\pm 1.18e-02}$	$4.136_{\pm 8.61e-01}$

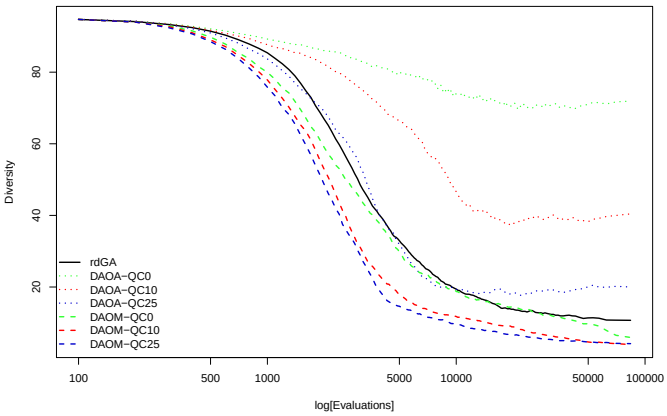
Table B.40: DAO-QC Convergence statistics for NK1-20-97 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.10: Convergence analysis DAO-QC *NK1-20-97*

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−0.121▽	−0.00685 -	0.0226▲	0.0481▲	0.0677▲	0.0624▲
DAOA-QC0		⁄	0.114▲	0.143▲	0.169▲	0.188▲	0.183▲
DAOA-QC10			⁄	0.0295▲	0.055▲	0.0745▲	0.0693▲
DAOA-QC25				⁄	0.0255▲	0.0451▲	0.0398▲
DAOM-QC0					⁄	0.0196▲	0.0143▲
DAOM-QC10						⁄	−0.00525 -
DAOM-QC25							⁄

Table B.41: DAO-QC Convergence statistics for NK2-20-97 Cross comparison Average fitness

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−0.079▽	−0.0133▽	0.00268 -	0.00395 -	0.0134▲	0.0114▲
DAOA-QC0		⁄	0.0657▲	0.0817▲	0.083▲	0.0924▲	0.0904▲
DAOA-QC10			⁄	0.016▲	0.0173▲	0.0267▲	0.0247▲
DAOA-QC25				⁄	0.00128 -	0.0107▲	0.00875 -
DAOM-QC0					⁄	0.00945▲	0.00747 -
DAOM-QC10						⁄	−0.00198 -
DAOM-QC25							⁄

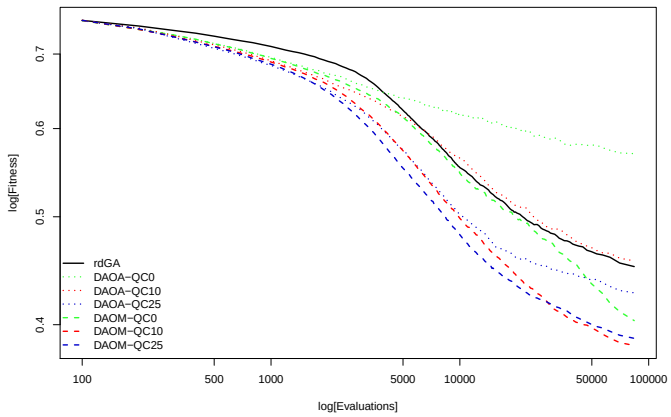
Table B.42: DAO-QC Convergence statistics for NK2-20-97 Cross comparison Best fitness

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−61.023▲	−28.110▲	−10.432▲	5.236▽	6.764▽	6.653▽
DAOA-QC0		⁄	32.913▽	50.591▽	66.259▽	67.787▽	67.677▽
DAOA-QC10			⁄	17.678▽	33.346▽	34.874▽	34.763▽
DAOA-QC25				⁄	15.668▽	17.196▽	17.085▽
DAOM-QC0					⁄	1.528▽	1.417▽
DAOM-QC10						⁄	−0.110 -
DAOM-QC25							⁄

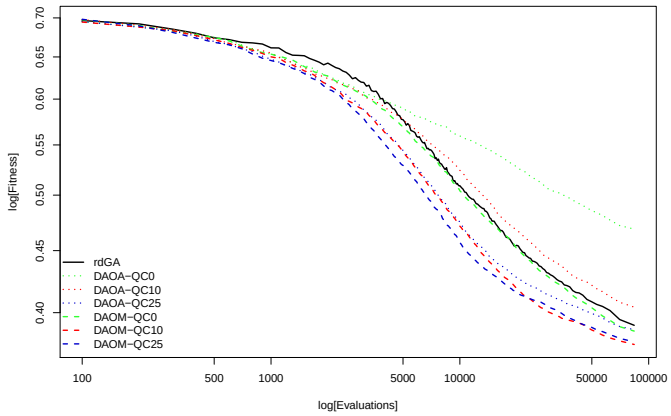
Table B.43: DAO-QC Convergence statistics for NK2-20-97 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.450_{\pm 1.03e-02}$	$0.389_{\pm 1.26e-02}$	$10.520_{\pm 1.04e+00}$
DAOA-QC0	$0.570_{\pm 1.21e-02}$	$0.468_{\pm 1.47e-02}$	$71.543_{\pm 3.14e+00}$
DAOA-QC10	$0.457_{\pm 1.84e-02}$	$0.403_{\pm 1.63e-02}$	$38.630_{\pm 5.97e+00}$
DAOA-QC25	$0.427_{\pm 1.57e-02}$	$0.387_{\pm 1.46e-02}$	$20.952_{\pm 4.44e+00}$
DAOM-QC0	$0.402_{\pm 1.86e-02}$	$0.385_{\pm 1.20e-02}$	$5.284_{\pm 1.46e+00}$
DAOM-QC10	$0.382_{\pm 1.52e-02}$	$0.376_{\pm 1.48e-02}$	$3.756_{\pm 8.72e-01}$
DAOM-QC25	$0.387_{\pm 1.47e-02}$	$0.378_{\pm 1.42e-02}$	$3.867_{\pm 7.60e-01}$

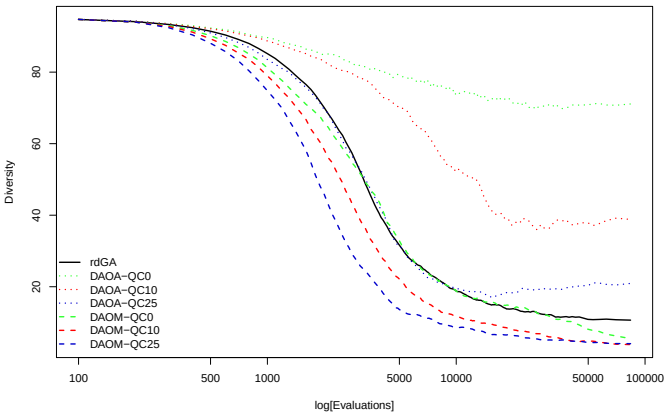
Table B.44: DAO-QC Convergence statistics for NK2-20-97 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.11: Convergence analysis DAO-QC *NK2-20-97*

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−0.132▽	−0.00272 -	0.0371▲	0.0452▲	0.0467▲	0.0499▲
DAOA-QC0		⁄	0.129▲	0.169▲	0.177▲	0.178▲	0.182▲
DAOA-QC10			⁄	0.0398▲	0.0479▲	0.0494▲	0.0526▲
DAOA-QC25				⁄	0.00812 -	0.00958 -	0.0128 -
DAOM-QC0					⁄	0.00145 -	0.0047 -
DAOM-QC10						⁄	0.00324 -
DAOM-QC25							⁄

Table B.45: DAO-QC Convergence statistics for NK1-4-67 Cross comparison Average fitness

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−0.0616▽	−0.0145 -	0.0102 -	−0.0065 -	−0.0149 -	−0.00676 -
DAOA-QC0		⁄	0.0471▲	0.0718▲	0.0551▲	0.0466▲	0.0548▲
DAOA-QC10			⁄	0.0247▲	0.00798 -	−0.00046 -	0.00772 -
DAOA-QC25				⁄	−0.0167▽	−0.0251▽	−0.0169▽
DAOM-QC0					⁄	−0.00844 -	−0.000263 -
DAOM-QC10						⁄	0.00818 -
DAOM-QC25							⁄

Table B.46: DAO-QC Convergence statistics for NK1-4-67 Cross comparison Best fitness

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−59.799▲	−39.409▲	−12.474▲	9.152▽	10.971▽	10.614▽
DAOA-QC0		⁄	20.390▽	47.325▽	68.950▽	70.770▽	70.413▽
DAOA-QC10			⁄	26.935▽	48.560▽	50.380▽	50.022▽
DAOA-QC25				⁄	21.625▽	23.445▽	23.087▽
DAOM-QC0					⁄	1.820▽	1.462▽
DAOM-QC10						⁄	−0.358 -
DAOM-QC25							⁄

Table B.47: DAO-QC Convergence statistics for NK1-4-67 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.355_{\pm 3.82e-02}$	$0.288_{\pm 3.98e-02}$	$16.650_{\pm 1.49e+00}$
DAOA-QC0	$0.486_{\pm 3.60e-02}$	$0.350_{\pm 3.76e-02}$	$76.449_{\pm 2.14e+00}$
DAOA-QC10	$0.357_{\pm 3.84e-02}$	$0.303_{\pm 3.85e-02}$	$56.058_{\pm 6.86e+00}$
DAOA-QC25	$0.317_{\pm 3.85e-02}$	$0.278_{\pm 3.82e-02}$	$29.123_{\pm 5.07e+00}$
DAOM-QC0	$0.309_{\pm 3.77e-02}$	$0.295_{\pm 3.62e-02}$	$7.498_{\pm 1.85e+00}$
DAOM-QC10	$0.308_{\pm 4.22e-02}$	$0.303_{\pm 4.19e-02}$	$5.678_{\pm 8.80e-01}$
DAOM-QC25	$0.305_{\pm 3.37e-02}$	$0.295_{\pm 3.38e-02}$	$6.036_{\pm 8.37e-01}$

Table B.48: DAO-QC Convergence statistics for NK1-4-67 Final values Average diversity

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−0.145▽	0.0148 -	0.0184▲	0.0301▲	0.0392▲	0.0383▲
DAOA-QC0		⁄	0.160▲	0.163▲	0.175▲	0.184▲	0.183▲
DAOA-QC10			⁄	0.00356 -	0.0153 -	0.0244▲	0.0235▲
DAOA-QC25				⁄	0.0117 -	0.0208▲	0.0199▲
DAOM-QC0					⁄	0.00913 -	0.00825 -
DAOM-QC10						⁄	−0.000883 -
DAOM-QC25							⁄

Table B.49: DAO-QC Convergence statistics for NK1-4-97 Cross comparison Average fitness

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−0.0699▽	0.00527 -	0.000733 -	0.000835 -	−0.00176 -	−0.000155 -
DAOA-QC0		⁄	0.0752▲	0.0707▲	0.0708▲	0.0682▲	0.0698▲
DAOA-QC10			⁄	−0.00453 -	−0.00443 -	−0.00702 -	−0.00542 -
DAOA-QC25				⁄	0.000101 -	−0.00249 -	−0.000889 -
DAOM-QC0					⁄	−0.00259 -	−0.00099 -
DAOM-QC10						⁄	0.0016 -
DAOM-QC25							⁄

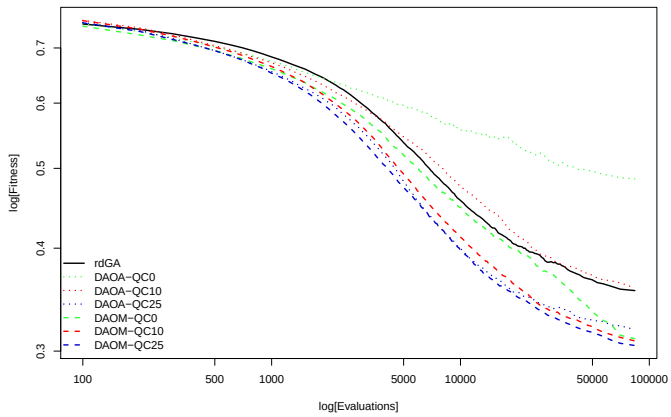
Table B.50: DAO-QC Convergence statistics for NK1-4-97 Cross comparison Best fitness

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−59.946▲	−29.413▲	−8.922▲	5.825▽	8.023▽	8.175▽
DAOA-QC0		⁄	30.533▽	51.024▽	65.771▽	67.969▽	68.121▽
DAOA-QC10			⁄	20.491▽	35.238▽	37.436▽	37.588▽
DAOA-QC25				⁄	14.746▽	16.944▽	17.096▽
DAOM-QC0					⁄	2.198▽	2.350▽
DAOM-QC10						⁄	0.152 -
DAOM-QC25							⁄

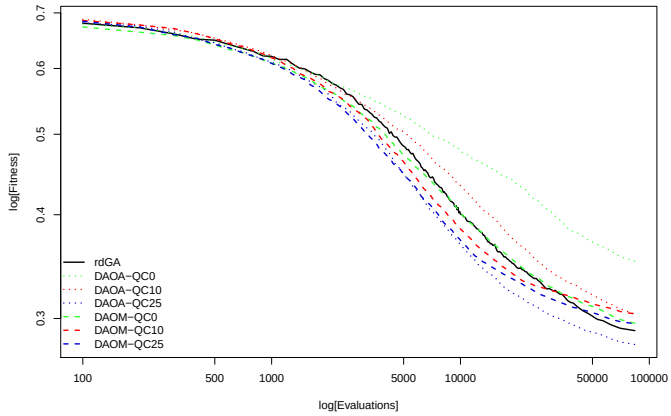
Table B.51: DAO-QC Convergence statistics for NK1-4-97 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.351_{\pm 2.50e-02}$	$0.305_{\pm 2.53e-02}$	$12.376_{\pm 1.06e+00}$
DAOA-QC0	$0.495_{\pm 3.08e-02}$	$0.375_{\pm 2.81e-02}$	$72.322_{\pm 2.58e+00}$
DAOA-QC10	$0.336_{\pm 3.70e-02}$	$0.300_{\pm 3.79e-02}$	$41.789_{\pm 4.90e+00}$
DAOA-QC25	$0.332_{\pm 3.11e-02}$	$0.304_{\pm 3.10e-02}$	$21.298_{\pm 3.47e+00}$
DAOM-QC0	$0.320_{\pm 3.74e-02}$	$0.304_{\pm 3.17e-02}$	$6.552_{\pm 2.59e+00}$
DAOM-QC10	$0.311_{\pm 2.60e-02}$	$0.307_{\pm 2.68e-02}$	$4.354_{\pm 7.47e-01}$
DAOM-QC25	$0.312_{\pm 3.89e-02}$	$0.305_{\pm 3.98e-02}$	$4.202_{\pm 6.49e-01}$

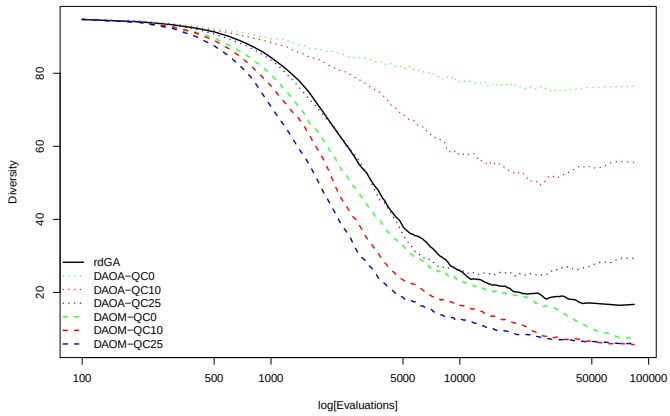
Table B.52: DAO-QC Convergence statistics for NK1-4-97 Final values Average diversity



(a) Average fitness

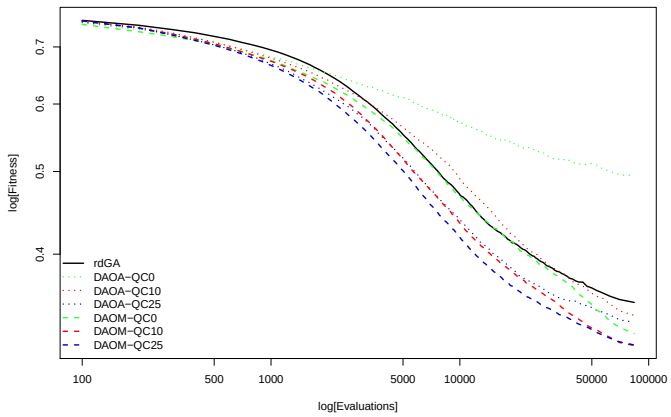


(b) Best fitness

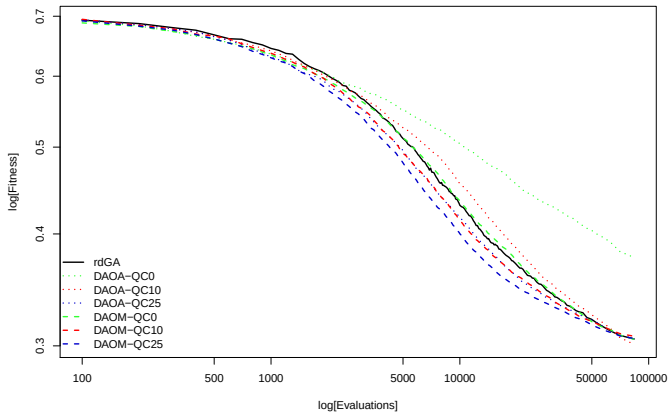


(c) Average diversity

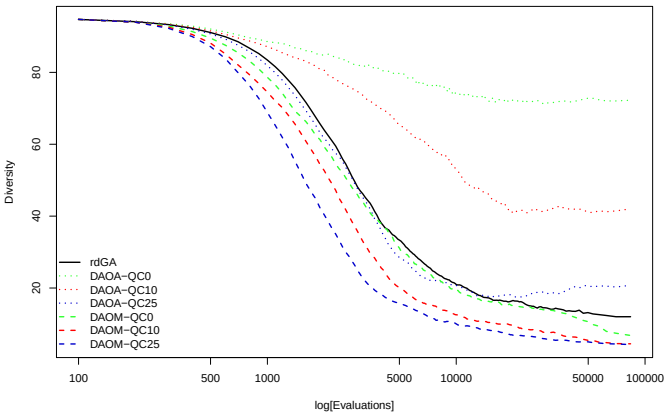
Figure B.12: Convergence analysis DAO-QC *NK1-4-67*



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.13: Convergence analysis DAO-QC *NK1-4-97*

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−0.136▽	0.000374 -	0.0294▲	0.0431▲	0.0667▲	0.0491▲
DAOA-QC0		⁄	0.137▲	0.166▲	0.179▲	0.203▲	0.185▲
DAOA-QC10			⁄	0.029▲	0.0427▲	0.0664▲	0.0487▲
DAOA-QC25				⁄	0.0137▲	0.0373▲	0.0197▲
DAOM-QC0					⁄	0.0237▲	0.00604 -
DAOM-QC10						⁄	−0.0176▽
DAOM-QC25							⁄

Table B.53: DAO-QC Convergence statistics for NK2-4-97 Cross comparison Average fitness

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−0.0798▽	−0.00789 -	0.0155▲	0.00872▲	0.0206▲	0.00559 -
DAOA-QC0		⁄	0.0719▲	0.0953▲	0.0885▲	0.100▲	0.0854▲
DAOA-QC10			⁄	0.0233▲	0.0166▲	0.0285▲	0.0135▲
DAOA-QC25				⁄	−0.00673 -	0.00518 -	−0.00986▽
DAOM-QC0					⁄	0.0119 -	−0.00313 -
DAOM-QC10						⁄	−0.015▽
DAOM-QC25							⁄

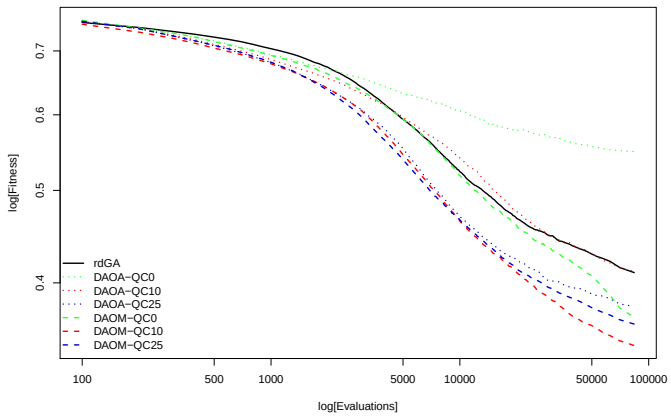
Table B.54: DAO-QC Convergence statistics for NK2-4-97 Cross comparison Best fitness

	rdGA	DAOA-QC0	DAOA-QC10	DAOA-QC25	DAOM-QC0	DAOM-QC10	DAOM-QC25
rdGA	⁄	−60.022▲	−28.592▲	−11.266▲	5.392▽	7.247▽	7.059▽
DAOA-QC0		⁄	31.430▽	48.756▽	65.413▽	67.269▽	67.080▽
DAOA-QC10			⁄	17.326▽	33.984▽	35.839▽	35.651▽
DAOA-QC25				⁄	16.658▽	18.513▽	18.325▽
DAOM-QC0					⁄	1.855▽	1.667▽
DAOM-QC10						⁄	−0.189 -
DAOM-QC25							⁄

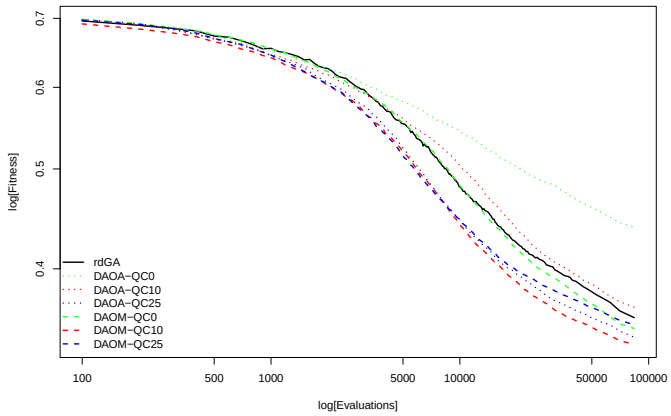
Table B.55: DAO-QC Convergence statistics for NK2-4-97 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.410_{\pm 1.61e-02}$	$0.358_{\pm 1.79e-02}$	$11.422_{\pm 8.10e-01}$
DAOA-QC0	$0.546_{\pm 1.97e-02}$	$0.438_{\pm 1.92e-02}$	$71.444_{\pm 2.81e+00}$
DAOA-QC10	$0.409_{\pm 2.32e-02}$	$0.366_{\pm 2.27e-02}$	$40.014_{\pm 6.61e+00}$
DAOA-QC25	$0.380_{\pm 1.73e-02}$	$0.342_{\pm 1.90e-02}$	$22.688_{\pm 3.17e+00}$
DAOM-QC0	$0.367_{\pm 2.54e-02}$	$0.349_{\pm 2.10e-02}$	$6.030_{\pm 2.30e+00}$
DAOM-QC10	$0.343_{\pm 2.37e-02}$	$0.337_{\pm 2.40e-02}$	$4.175_{\pm 8.54e-01}$
DAOM-QC25	$0.361_{\pm 1.50e-02}$	$0.352_{\pm 1.52e-02}$	$4.363_{\pm 6.59e-01}$

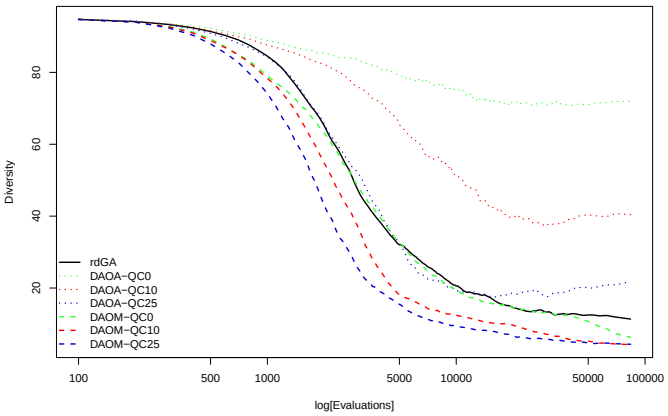
Table B.56: DAO-QC Convergence statistics for NK2-4-97 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.14: Convergence analysis DAO-QC *NK2-4-97*

B.4 Preference Based Genetic Algorithm Study

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	∕	−0.0291▽	0.0285▲	0.0747▲	0.111▲	0.103▲	0.0888▲
PBGA 0.3 0.7		∕	0.0576▲	0.104▲	0.140▲	0.132▲	0.118▲
PBGA 0.5 0.5			∕	0.0462▲	0.0828▲	0.0743▲	0.0603▲
PBGA 0.7 0.3				∕	0.0366▲	0.0281▲	0.0141▲
PBGA 0.8 0.2					∕	−0.00852▽	−0.0225▽
PBGA 0.9 0.1						∕	−0.014▽
PBGA 1.0 0.0							∕

Table B.57: PBGA Convergence statistics for NK1-20-67 Cross comparison Average fitness

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	∕	−0.0209▽	0.0197▲	0.0426▲	0.0458▲	0.0288▲	0.0142▲
PBGA 0.3 0.7		∕	0.0406▲	0.0635▲	0.0666▲	0.0497▲	0.035▲
PBGA 0.5 0.5			∕	0.0229▲	0.026▲	0.00909▲	−0.00556 -
PBGA 0.7 0.3				∕	0.00313 -	−0.0138▽	−0.0285▽
PBGA 0.8 0.2					∕	−0.0169▽	−0.0316▽
PBGA 0.9 0.1						∕	−0.0147▽
PBGA 1.0 0.0							∕

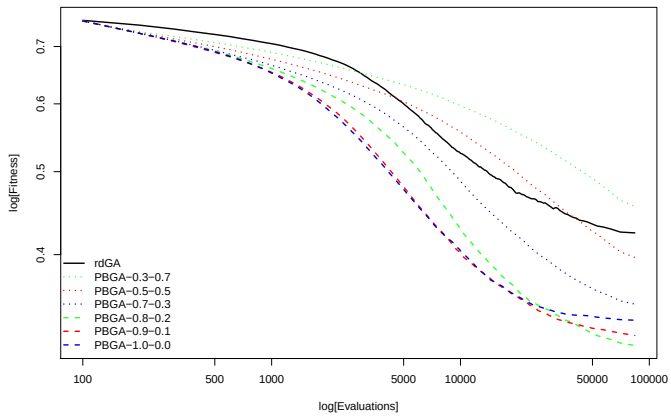
Table B.58: PBGA Convergence statistics for NK1-20-67 Cross comparison Best fitness

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	∕	−73.954▲	−61.736▲	−33.614▲	−0.746 -	12.850▽	14.874▽
PBGA 0.3 0.7		∕	12.218▽	40.340▽	73.208▽	86.804▽	88.828▽
PBGA 0.5 0.5			∕	28.122▽	60.990▽	74.586▽	76.610▽
PBGA 0.7 0.3				∕	32.867▽	46.464▽	48.488▽
PBGA 0.8 0.2					∕	13.597▽	15.620▽
PBGA 0.9 0.1						∕	2.024▽
PBGA 1.0 0.0							∕

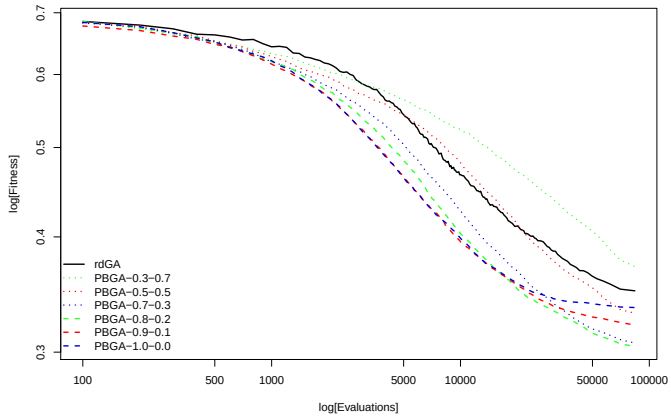
Table B.59: PBGA Convergence statistics for NK1-20-67 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.424_{\pm 1.61e-02}$	$0.350_{\pm 1.73e-02}$	$14.874_{\pm 1.81e+00}$
PBGA 0.3 0.7	$0.453_{\pm 6.19e-03}$	$0.370_{\pm 1.36e-02}$	$88.828_{\pm 3.64e-01}$
PBGA 0.5 0.5	$0.396_{\pm 1.14e-02}$	$0.330_{\pm 1.38e-02}$	$76.610_{\pm 1.28e+00}$
PBGA 0.7 0.3	$0.349_{\pm 1.44e-02}$	$0.307_{\pm 1.47e-02}$	$48.488_{\pm 4.85e+00}$
PBGA 0.8 0.2	$0.313_{\pm 1.48e-02}$	$0.304_{\pm 1.39e-02}$	$15.620_{\pm 4.84e+00}$
PBGA 0.9 0.1	$0.321_{\pm 1.30e-02}$	$0.321_{\pm 1.28e-02}$	$2.024_{\pm 1.66e+00}$
PBGA 1.0 0.0	$0.335_{\pm 1.59e-02}$	$0.335_{\pm 1.59e-02}$	$0.000_{\pm 0.00e+00}$

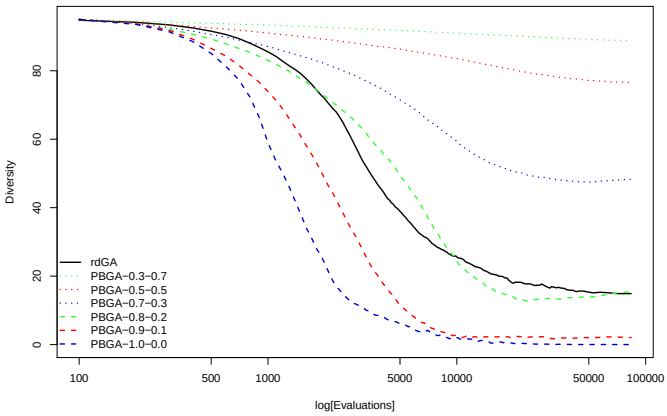
Table B.60: PBGA Convergence statistics for NK1-20-67 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.15: Convergence analysis PBGA *NK1-20-67*

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	⁄	−0.0115▽	0.0409▲	0.0862▲	0.119▲	0.118▲	0.106▲
PBGA 0.3 0.7		⁄	0.0524▲	0.0977▲	0.131▲	0.129▲	0.117▲
PBGA 0.5 0.5			⁄	0.0453▲	0.0785▲	0.077▲	0.065▲
PBGA 0.7 0.3				⁄	0.0332▲	0.0317▲	0.0197▲
PBGA 0.8 0.2					⁄	−0.00153 -	−0.0135▽
PBGA 0.9 0.1						⁄	−0.0119▽
PBGA 1.0 0.0							⁄

Table B.61: PBGA Convergence statistics for NK2-20-67 Cross comparison Average fitness

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	⁄	−0.0188▽	0.0175▲	0.0414▲	0.0435▲	0.0323▲	0.0199▲
PBGA 0.3 0.7		⁄	0.0363▲	0.0601▲	0.0623▲	0.0511▲	0.0386▲
PBGA 0.5 0.5			⁄	0.0239▲	0.0261▲	0.0149▲	0.00237 -
PBGA 0.7 0.3				⁄	0.00218 -	−0.00901 -	−0.0215▽
PBGA 0.8 0.2					⁄	−0.0112▽	−0.0237▽
PBGA 0.9 0.1						⁄	−0.0125▽
PBGA 1.0 0.0							⁄

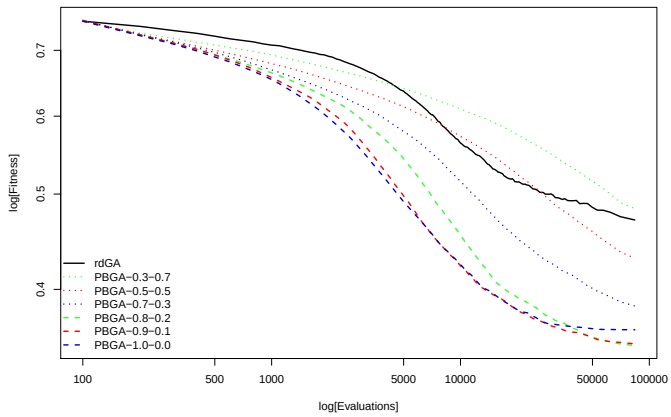
Table B.62: PBGA Convergence statistics for NK2-20-67 Cross comparison Best fitness

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	⁄	−74.386▲	−63.096▲	−37.065▲	−1.023 -	13.011▽	14.748▽
PBGA 0.3 0.7		⁄	11.289▽	37.321▽	73.362▽	87.396▽	89.133▽
PBGA 0.5 0.5			⁄	26.031▽	62.073▽	76.107▽	77.844▽
PBGA 0.7 0.3				⁄	36.042▽	50.075▽	51.813▽
PBGA 0.8 0.2					⁄	14.034▽	15.771▽
PBGA 0.9 0.1						⁄	1.737▽
PBGA 1.0 0.0							⁄

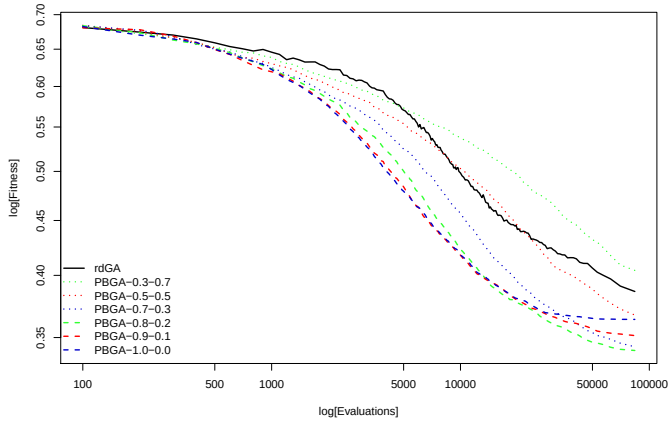
Table B.63: PBGA Convergence statistics for NK2-20-67 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.470_{\pm 1.37e-02}$	$0.384_{\pm 1.98e-02}$	$14.748_{\pm 1.63e+00}$
PBGA 0.3 0.7	$0.481_{\pm 7.29e-03}$	$0.403_{\pm 1.55e-02}$	$89.133_{\pm 3.62e-01}$
PBGA 0.5 0.5	$0.429_{\pm 7.70e-03}$	$0.366_{\pm 1.30e-02}$	$77.844_{\pm 9.47e-01}$
PBGA 0.7 0.3	$0.384_{\pm 1.50e-02}$	$0.342_{\pm 1.69e-02}$	$51.813_{\pm 3.39e+00}$
PBGA 0.8 0.2	$0.350_{\pm 1.74e-02}$	$0.340_{\pm 1.28e-02}$	$15.771_{\pm 8.51e+00}$
PBGA 0.9 0.1	$0.352_{\pm 1.36e-02}$	$0.351_{\pm 1.35e-02}$	$1.737_{\pm 1.39e+00}$
PBGA 1.0 0.0	$0.364_{\pm 1.89e-02}$	$0.364_{\pm 1.89e-02}$	$0.000_{\pm 0.00e+00}$

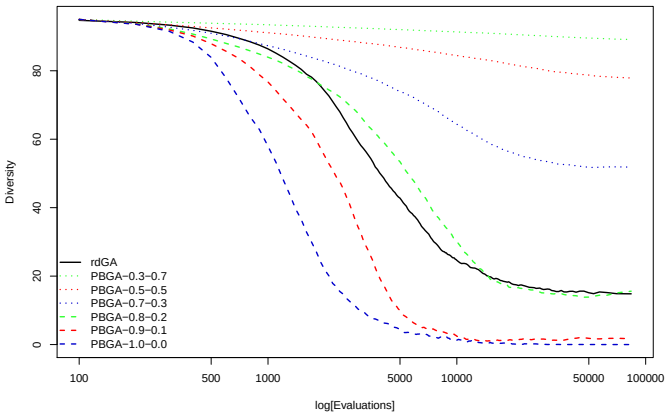
Table B.64: PBGA Convergence statistics for NK2-20-67 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.16: Convergence analysis PBGA *NK2-20-67*

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	⁄	−0.0622▽	0.0179 -	0.076▲	0.091▲	0.0773▲	0.0553▲
PBGA 0.3 0.7		⁄	0.0801▲	0.138▲	0.153▲	0.139▲	0.118▲
PBGA 0.5 0.5			⁄	0.058▲	0.0731▲	0.0594▲	0.0374▲
PBGA 0.7 0.3				⁄	0.0151 -	0.00133 -	−0.0206▽
PBGA 0.8 0.2					⁄	−0.0137 -	−0.0357▽
PBGA 0.9 0.1						⁄	−0.0219▽
PBGA 1.0 0.0							⁄

Table B.65: PBGA Convergence statistics for NK1-4-67 Cross comparison Average fitness

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	⁄	−0.0465▽	0.0129 -	0.0381▲	0.0341▲	0.0129 -	−0.0107 -
PBGA 0.3 0.7		⁄	0.0594▲	0.0846▲	0.0806▲	0.0594▲	0.0358▲
PBGA 0.5 0.5			⁄	0.0252▲	0.0212▲	$2.23e-05$ -	−0.0236 -
PBGA 0.7 0.3				⁄	−0.004 -	−0.0252▽	−0.0488▽
PBGA 0.8 0.2					⁄	−0.0212▽	−0.0448▽
PBGA 0.9 0.1						⁄	−0.0236▽
PBGA 1.0 0.0							⁄

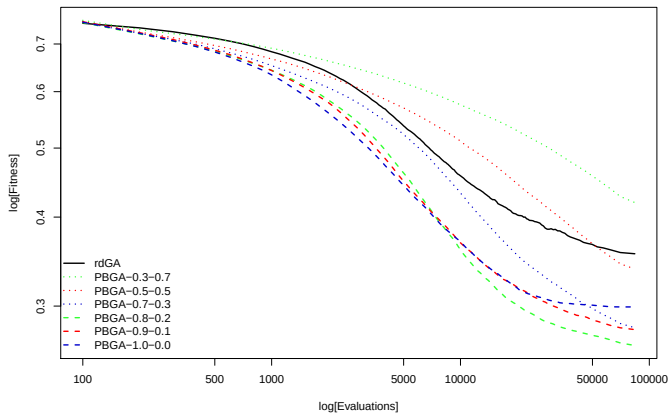
Table B.66: PBGA Convergence statistics for NK1-4-67 Cross comparison Best fitness

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	⁄	−71.470▲	−59.068▲	−32.502▲	−12.689▲	8.723▽	16.215▽
PBGA 0.3 0.7		⁄	12.402▽	38.969▽	58.781▽	80.194▽	87.686▽
PBGA 0.5 0.5			⁄	26.567▽	46.379▽	67.792▽	75.284▽
PBGA 0.7 0.3				⁄	19.813▽	41.225▽	48.717▽
PBGA 0.8 0.2					⁄	21.413▽	28.905▽
PBGA 0.9 0.1						⁄	7.492▽
PBGA 1.0 0.0							⁄

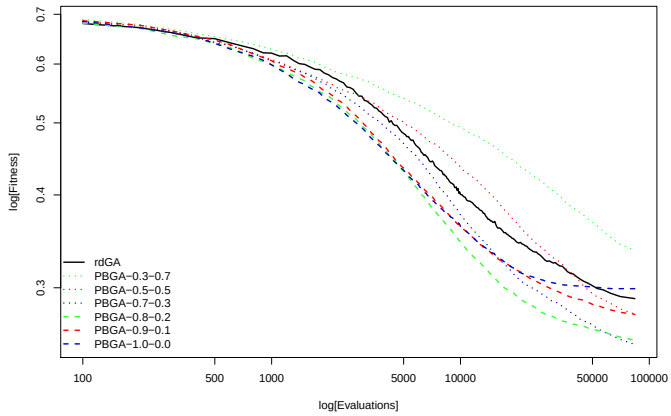
Table B.67: PBGA Convergence statistics for NK1-4-67 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.355_{\pm 3.82e-02}$	$0.288_{\pm 3.98e-02}$	$16.650_{\pm 1.49e+00}$
PBGA 0.3 0.7	$0.417_{\pm 3.72e-02}$	$0.335_{\pm 3.78e-02}$	$88.120_{\pm 7.38e-01}$
PBGA 0.5 0.5	$0.337_{\pm 4.08e-02}$	$0.276_{\pm 4.13e-02}$	$75.718_{\pm 2.07e+00}$
PBGA 0.7 0.3	$0.279_{\pm 3.63e-02}$	$0.250_{\pm 3.68e-02}$	$49.152_{\pm 4.68e+00}$
PBGA 0.8 0.2	$0.263_{\pm 3.78e-02}$	$0.254_{\pm 3.77e-02}$	$29.339_{\pm 4.84e+00}$
PBGA 0.9 0.1	$0.277_{\pm 3.25e-02}$	$0.276_{\pm 3.22e-02}$	$7.926_{\pm 3.34e+00}$
PBGA 1.0 0.0	$0.299_{\pm 3.32e-02}$	$0.299_{\pm 3.32e-02}$	$0.434_{\pm 4.00e-01}$

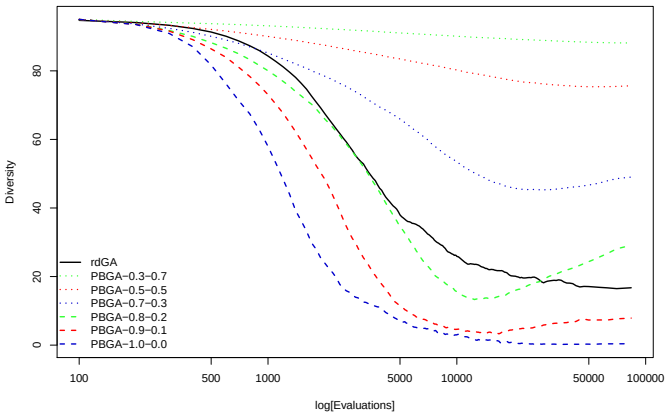
Table B.68: PBGA Convergence statistics for NK1-4-67 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.17: Convergence analysis PBGA *NK1-4-67*

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	⁄	−0.0732▽	−0.00268 -	0.053▲	0.0907▲	0.0899▲	0.0768▲
PBGA 0.3 0.7		⁄	0.0705▲	0.126▲	0.164▲	0.163▲	0.150▲
PBGA 0.5 0.5			⁄	0.0556▲	0.0934▲	0.0926▲	0.0795▲
PBGA 0.7 0.3				⁄	0.0378▲	0.037▲	0.0238▲
PBGA 0.8 0.2					⁄	−0.000808 -	−0.0139▽
PBGA 0.9 0.1						⁄	−0.0131▽
PBGA 1.0 0.0							⁄

Table B.69: PBGA Convergence statistics for NK1-20-97 Cross comparison Average fitness

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	⁄	−0.0365▽	0.0171▲	0.0469▲	0.0477▲	0.0391▲	0.0253▲
PBGA 0.3 0.7		⁄	0.0536▲	0.0834▲	0.0842▲	0.0756▲	0.0618▲
PBGA 0.5 0.5			⁄	0.0298▲	0.0306▲	0.022▲	0.00821▲
PBGA 0.7 0.3				⁄	0.000796 -	−0.00781▽	−0.0216▽
PBGA 0.8 0.2					⁄	−0.0086▽	−0.0224▽
PBGA 0.9 0.1						⁄	−0.0138▽
PBGA 1.0 0.0							⁄

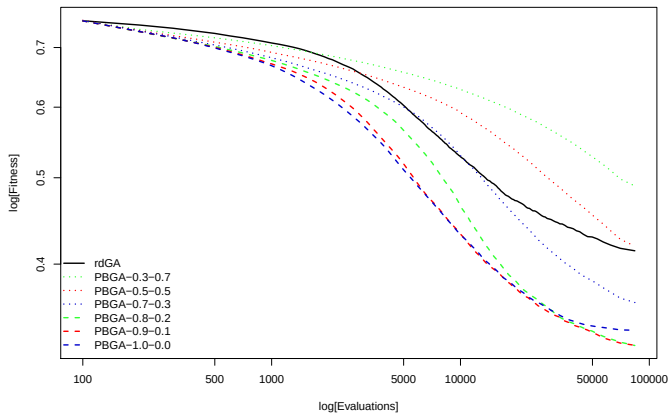
Table B.70: PBGA Convergence statistics for NK1-20-97 Cross comparison Best fitness

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	⁄	−78.385▲	−65.622▲	−36.062▲	−3.633▲	8.404▽	10.684▽
PBGA 0.3 0.7		⁄	12.763▽	42.323▽	74.752▽	86.788▽	89.069▽
PBGA 0.5 0.5			⁄	29.560▽	61.989▽	74.026▽	76.306▽
PBGA 0.7 0.3				⁄	32.429▽	44.466▽	46.746▽
PBGA 0.8 0.2					⁄	12.036▽	14.317▽
PBGA 0.9 0.1						⁄	2.281▽
PBGA 1.0 0.0							⁄

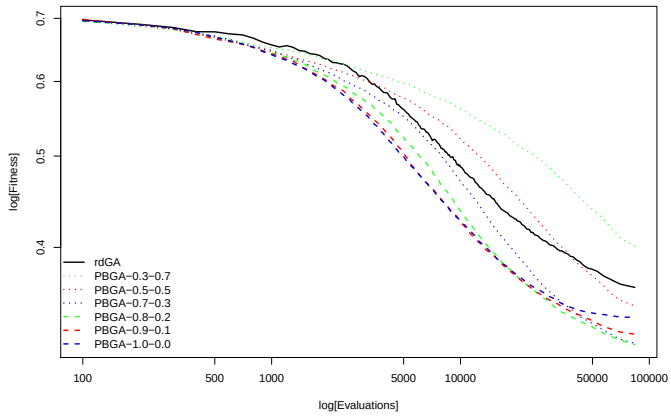
Table B.71: PBGA Convergence statistics for NK1-20-97 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.414_{\pm 1.10e-02}$	$0.362_{\pm 1.31e-02}$	$10.703_{\pm 1.28e+00}$
PBGA 0.3 0.7	$0.487_{\pm 7.79e-03}$	$0.399_{\pm 1.32e-02}$	$89.087_{\pm 3.15e-01}$
PBGA 0.5 0.5	$0.417_{\pm 9.75e-03}$	$0.345_{\pm 1.21e-02}$	$76.325_{\pm 1.00e+00}$
PBGA 0.7 0.3	$0.361_{\pm 1.15e-02}$	$0.315_{\pm 1.20e-02}$	$46.765_{\pm 3.72e+00}$
PBGA 0.8 0.2	$0.323_{\pm 1.20e-02}$	$0.315_{\pm 1.14e-02}$	$14.335_{\pm 3.82e+00}$
PBGA 0.9 0.1	$0.324_{\pm 1.33e-02}$	$0.323_{\pm 1.32e-02}$	$2.299_{\pm 1.61e+00}$
PBGA 1.0 0.0	$0.337_{\pm 1.61e-02}$	$0.337_{\pm 1.62e-02}$	$0.0182_{\pm 9.97e-02}$

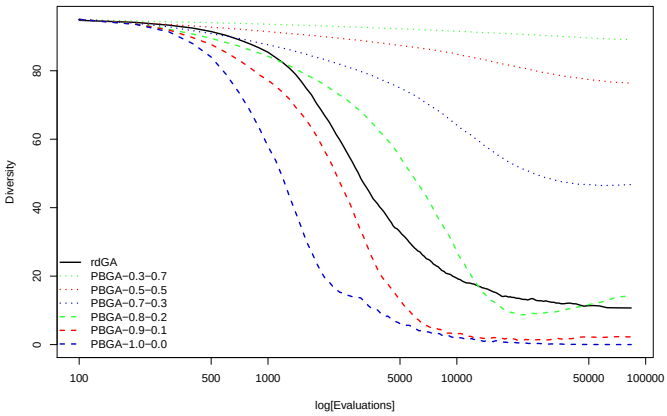
Table B.72: PBGA Convergence statistics for NK1-20-97 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.18: Convergence analysis PBGA *NK1-20-97*

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	／	−0.0571▽	0.00483 -	0.0542▲	0.0987▲	0.0935▲	0.0855▲
PBGA 0.3 0.7		／	0.0619▲	0.111▲	0.156▲	0.151▲	0.143▲
PBGA 0.5 0.5			／	0.0493▲	0.0938▲	0.0887▲	0.0807▲
PBGA 0.7 0.3				／	0.0445▲	0.0393▲	0.0313▲
PBGA 0.8 0.2					／	−0.00518 -	−0.0132▽
PBGA 0.9 0.1						／	−0.00799▽
PBGA 1.0 0.0							／

Table B.73: PBGA Convergence statistics for NK2-20-97 Cross comparison Average fitness

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	／	−0.0325▽	0.0122▲	0.0409▲	0.0459▲	0.0338▲	0.0251▲
PBGA 0.3 0.7		／	0.0446▲	0.0734▲	0.0784▲	0.0662▲	0.0575▲
PBGA 0.5 0.5			／	0.0288▲	0.0337▲	0.0216▲	0.0129▲
PBGA 0.7 0.3				／	0.00494 -	−0.0072▽	−0.0159▽
PBGA 0.8 0.2					／	−0.0121▽	−0.0208▽
PBGA 0.9 0.1						／	−0.00868▽
PBGA 1.0 0.0							／

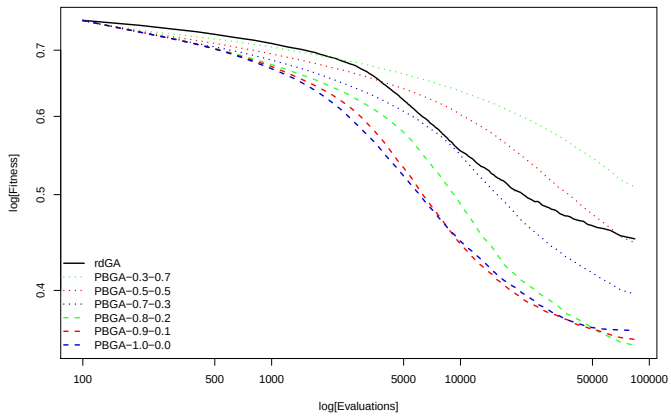
Table B.74: PBGA Convergence statistics for NK2-20-97 Cross comparison Best fitness

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	／	−78.755▲	−66.915▲	−40.115▲	−0.650 -	8.795▽	10.520▽
PBGA 0.3 0.7		／	11.840▽	38.640▽	78.106▽	87.550▽	89.275▽
PBGA 0.5 0.5			／	26.800▽	66.265▽	75.710▽	77.435▽
PBGA 0.7 0.3				／	39.465▽	48.910▽	50.635▽
PBGA 0.8 0.2					／	9.444▽	11.170▽
PBGA 0.9 0.1						／	1.725▽
PBGA 1.0 0.0							／

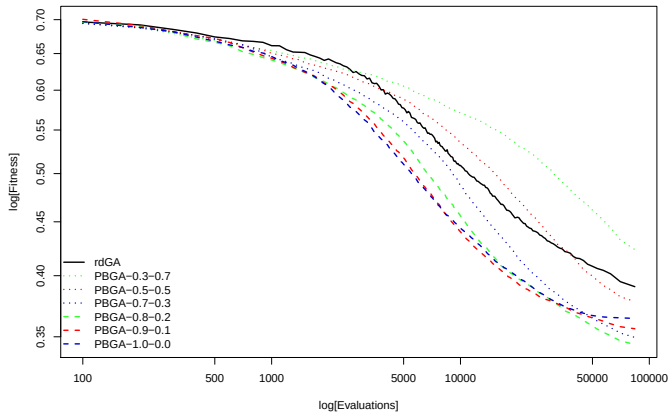
Table B.75: PBGA Convergence statistics for NK2-20-97 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.450_{\pm 1.03e-02}$	$0.389_{\pm 1.26e-02}$	$10.520_{\pm 1.04e+00}$
PBGA 0.3 0.7	$0.507_{\pm 6.09e-03}$	$0.422_{\pm 1.15e-02}$	$89.275_{\pm 2.97e-01}$
PBGA 0.5 0.5	$0.445_{\pm 8.72e-03}$	$0.377_{\pm 1.47e-02}$	$77.435_{\pm 1.15e+00}$
PBGA 0.7 0.3	$0.396_{\pm 1.10e-02}$	$0.348_{\pm 1.13e-02}$	$50.635_{\pm 3.13e+00}$
PBGA 0.8 0.2	$0.351_{\pm 1.44e-02}$	$0.343_{\pm 1.24e-02}$	$11.170_{\pm 4.04e+00}$
PBGA 0.9 0.1	$0.356_{\pm 1.13e-02}$	$0.355_{\pm 1.14e-02}$	$1.725_{\pm 1.39e+00}$
PBGA 1.0 0.0	$0.364_{\pm 1.21e-02}$	$0.364_{\pm 1.21e-02}$	$0.000_{\pm 0.00e+00}$

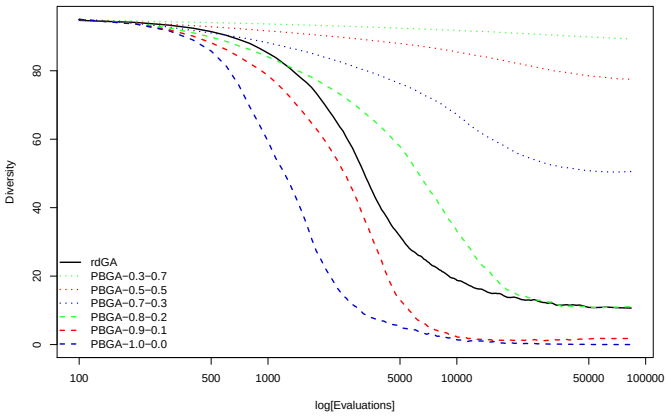
Table B.76: PBGA Convergence statistics for NK2-20-97 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.19: Convergence analysis PBGA *NK2-20-97*

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	⁄	−0.0926▽	−0.0197▽	0.0629▲	0.0755▲	0.0612▲	0.0525▲
PBGA 0.3 0.7		⁄	0.073▲	0.156▲	0.168▲	0.154▲	0.145▲
PBGA 0.5 0.5			⁄	0.0826▲	0.0952▲	0.0809▲	0.0722▲
PBGA 0.7 0.3				⁄	0.0127 -	−0.00168 -	−0.0104 -
PBGA 0.8 0.2					⁄	−0.0144 -	−0.023▽
PBGA 0.9 0.1						⁄	−0.00868 -
PBGA 1.0 0.0							⁄

Table B.77: PBGA Convergence statistics for NK1-4-97 Cross comparison Average fitness

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	⁄	−0.0487▽	−0.000144 -	0.0449▲	0.040▲	0.0176▲	0.00716 -
PBGA 0.3 0.7		⁄	0.0486▲	0.0936▲	0.0887▲	0.0663▲	0.0559▲
PBGA 0.5 0.5			⁄	0.045▲	0.0401▲	0.0177 -	0.0073 -
PBGA 0.7 0.3				⁄	−0.00487 -	−0.0273▽	−0.0377▽
PBGA 0.8 0.2					⁄	−0.0224▽	−0.0328▽
PBGA 0.9 0.1						⁄	−0.0104 -
PBGA 1.0 0.0							⁄

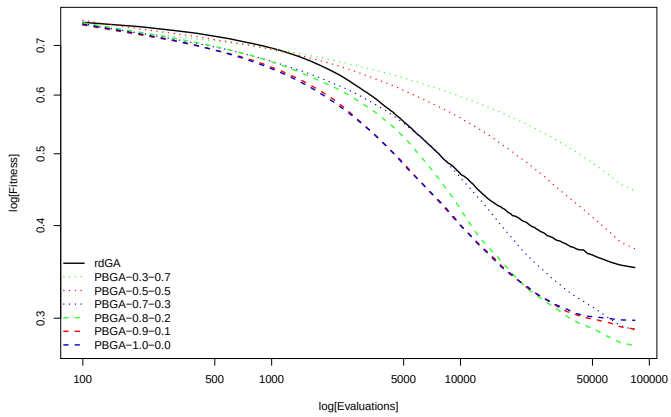
Table B.78: PBGA Convergence statistics for NK1-4-97 Cross comparison Best fitness

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	⁄	−76.048▲	−62.779▲	−34.313▲	−12.146▲	5.150▽	12.179▽
PBGA 0.3 0.7		⁄	13.269▽	41.735▽	63.902▽	81.198▽	88.227▽
PBGA 0.5 0.5			⁄	28.466▽	50.633▽	67.929▽	74.958▽
PBGA 0.7 0.3				⁄	22.167▽	39.463▽	46.492▽
PBGA 0.8 0.2					⁄	17.296▽	24.325▽
PBGA 0.9 0.1						⁄	7.029▽
PBGA 1.0 0.0							⁄

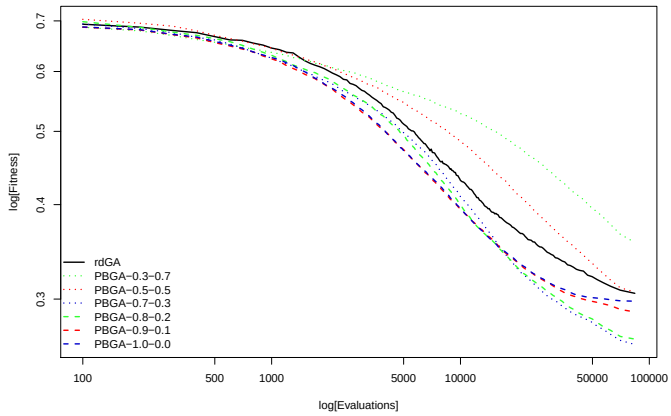
Table B.79: PBGA Convergence statistics for NK1-4-97 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.351_{\pm 2.50e-02}$	$0.305_{\pm 2.53e-02}$	$12.376_{\pm 1.06e+00}$
PBGA 0.3 0.7	$0.443_{\pm 2.93e-02}$	$0.354_{\pm 3.15e-02}$	$88.424_{\pm 7.00e-01}$
PBGA 0.5 0.5	$0.370_{\pm 3.44e-02}$	$0.305_{\pm 3.69e-02}$	$75.155_{\pm 1.71e+00}$
PBGA 0.7 0.3	$0.288_{\pm 3.88e-02}$	$0.260_{\pm 3.68e-02}$	$46.689_{\pm 4.26e+00}$
PBGA 0.8 0.2	$0.275_{\pm 3.00e-02}$	$0.265_{\pm 2.88e-02}$	$24.522_{\pm 3.90e+00}$
PBGA 0.9 0.1	$0.289_{\pm 3.24e-02}$	$0.288_{\pm 3.23e-02}$	$7.226_{\pm 2.52e+00}$
PBGA 1.0 0.0	$0.298_{\pm 2.59e-02}$	$0.298_{\pm 2.59e-02}$	$0.197_{\pm 2.69e-01}$

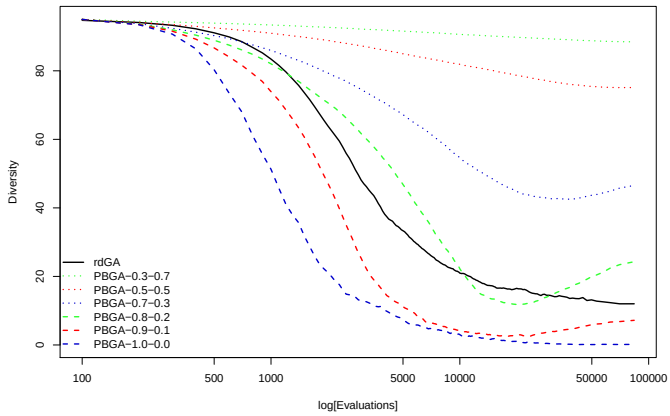
Table B.80: PBGA Convergence statistics for NK1-4-97 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.20: Convergence analysis PBGA *NK1-4-97*

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	⁄	−0.0727▽	−0.00643 -	0.0524▲	0.0917▲	0.0811▲	0.073▲
PBGA 0.3 0.7		⁄	0.0663▲	0.125▲	0.164▲	0.154▲	0.146▲
PBGA 0.5 0.5			⁄	0.0588▲	0.0982▲	0.0876▲	0.0794▲
PBGA 0.7 0.3				⁄	0.0394▲	0.0288▲	0.0206▲
PBGA 0.8 0.2					⁄	−0.0106▽	−0.0188▽
PBGA 0.9 0.1						⁄	−0.00818 -
PBGA 1.0 0.0							⁄

Table B.81: PBGA Convergence statistics for NK2-4-97 Cross comparison Average fitness

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	⁄	−0.0351▽	0.012▲	0.0405▲	0.0483▲	0.0303▲	0.0211▲
PBGA 0.3 0.7		⁄	0.0471▲	0.0756▲	0.0834▲	0.0654▲	0.0561▲
PBGA 0.5 0.5			⁄	0.0285▲	0.0363▲	0.0183▲	0.00904 -
PBGA 0.7 0.3				⁄	0.00781 -	−0.0102 -	−0.0195▽
PBGA 0.8 0.2					⁄	−0.018▽	−0.0273▽
PBGA 0.9 0.1						⁄	−0.00929 -
PBGA 1.0 0.0							⁄

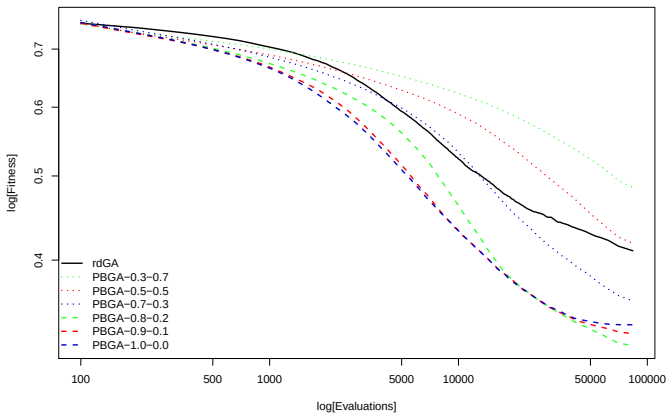
Table B.82: PBGA Convergence statistics for NK2-4-97 Cross comparison Best fitness

	rdGA	PBGA 0.3 0.7	PBGA 0.5 0.5	PBGA 0.7 0.3	PBGA 0.8 0.2	PBGA 0.9 0.1	PBGA 1.0 0.0
rdGA	⁄	−77.624▲	−65.439▲	−35.280▲	−6.687▲	6.423▽	11.292▽
PBGA 0.3 0.7		⁄	12.185▽	42.344▽	70.938▽	84.047▽	88.916▽
PBGA 0.5 0.5			⁄	30.160▽	58.753▽	71.862▽	76.731▽
PBGA 0.7 0.3				⁄	28.593▽	41.703▽	46.571▽
PBGA 0.8 0.2					⁄	13.109▽	17.978▽
PBGA 0.9 0.1						⁄	4.869▽
PBGA 1.0 0.0							⁄

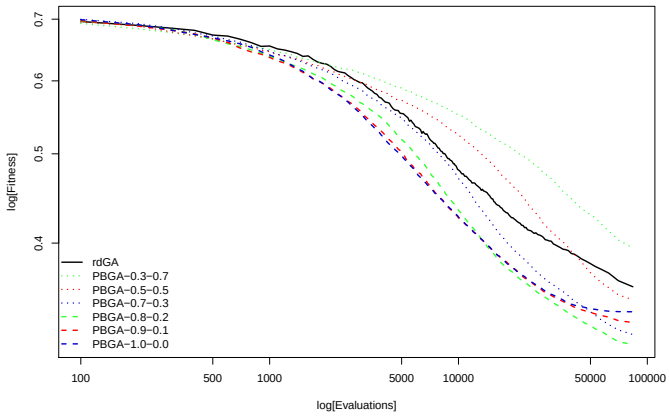
Table B.83: PBGA Convergence statistics for NK2-4-97 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.410_{\pm 1.61e-02}$	$0.358_{\pm 1.79e-02}$	$11.422_{\pm 8.10e-01}$
PBGA 0.3 0.7	$0.483_{\pm 1.33e-02}$	$0.393_{\pm 1.88e-02}$	$89.046_{\pm 3.22e-01}$
PBGA 0.5 0.5	$0.416_{\pm 1.79e-02}$	$0.346_{\pm 2.12e-02}$	$76.861_{\pm 1.30e+00}$
PBGA 0.7 0.3	$0.357_{\pm 2.15e-02}$	$0.317_{\pm 2.27e-02}$	$46.702_{\pm 4.44e+00}$
PBGA 0.8 0.2	$0.318_{\pm 2.05e-02}$	$0.309_{\pm 2.05e-02}$	$18.108_{\pm 3.65e+00}$
PBGA 0.9 0.1	$0.329_{\pm 1.77e-02}$	$0.327_{\pm 1.76e-02}$	$4.999_{\pm 2.10e+00}$
PBGA 1.0 0.0	$0.337_{\pm 2.41e-02}$	$0.337_{\pm 2.41e-02}$	$0.130_{\pm 2.12e-01}$

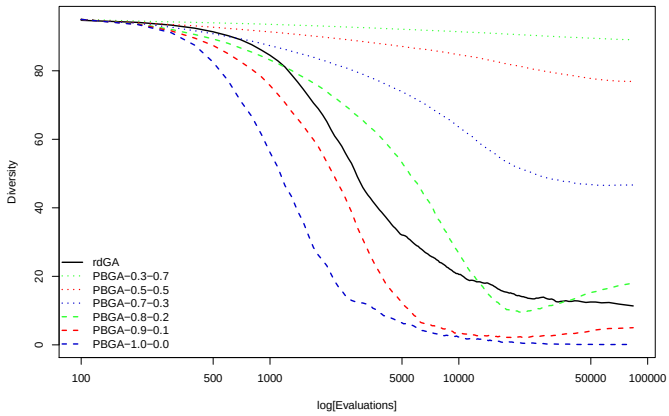
Table B.84: PBGA Convergence statistics for NK2-4-97 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.21: Convergence analysis PBGA *NK2-4-97*

B.5 Summary of Algorithm Experiments Study

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	−0.00988▽	0.0287▲	0.076▲	0.0697▲	0.111▲	0.103▲
DAOA-QC10		⁄	0.0386▲	0.0859▲	0.0796▲	0.121▲	0.113▲
DAOA-QC25			⁄	0.0473▲	0.041▲	0.0826▲	0.0741▲
DAOM-QC10				⁄	−0.00629▽	0.0353▲	0.0268▲
DAOM-QC25					⁄	0.0416▲	0.0331▲
PBGA 0.8 0.2						⁄	−0.00852▽
PBGA 0.9 0.1							⁄

Table B.85: DAO vs. PBGA Convergence statistics for NK1-20-67 Cross comparison Average fitness

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	−0.0198▽	0.000614 -	0.0062 -	0.00703 -	0.0458▲	0.0288▲
DAOA-QC10		⁄	0.0204▲	0.026▲	0.0268▲	0.0655▲	0.0486▲
DAOA-QC25			⁄	0.00559 -	0.00642 -	0.0451▲	0.0282▲
DAOM-QC10				⁄	0.000828 -	0.0395▲	0.0226▲
DAOM-QC25					⁄	0.0387▲	0.0218▲
PBGA 0.8 0.2						⁄	−0.0169▽
PBGA 0.9 0.1							⁄

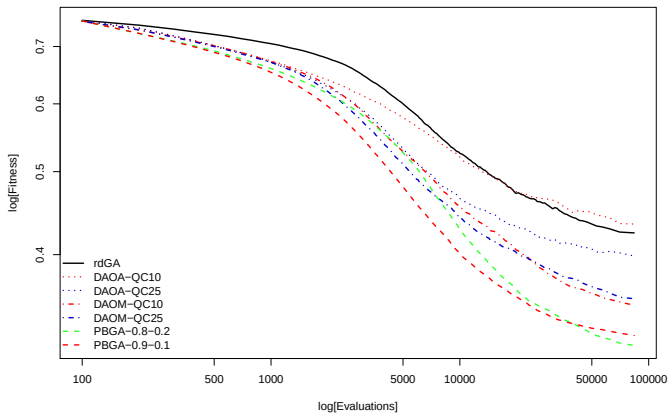
Table B.86: DAO vs. PBGA Convergence statistics for NK1-20-67 Cross comparison Best fitness

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	−37.734▲	−16.130▲	10.031▽	9.291▽	−0.746 -	12.850▽
DAOA-QC10		⁄	21.604▽	47.765▽	47.025▽	36.987▽	50.584▽
DAOA-QC25			⁄	26.161▽	25.421▽	15.384▽	28.980▽
DAOM-QC10				⁄	−0.740▲	−10.777▲	2.819▽
DAOM-QC25					⁄	−10.037▲	3.559▽
PBGA 0.8 0.2						⁄	13.597▽
PBGA 0.9 0.1							⁄

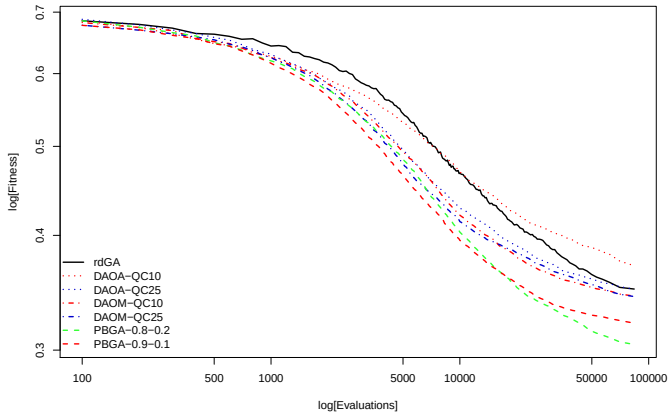
Table B.87: DAO vs. PBGA Convergence statistics for NK1-20-67 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.424_{\pm 1.61e-02}$	$0.350_{\pm 1.73e-02}$	$14.874_{\pm 1.81e+00}$
DAOA-QC10	$0.434_{\pm 1.73e-02}$	$0.369_{\pm 1.79e-02}$	$52.608_{\pm 7.00e+00}$
DAOA-QC25	$0.395_{\pm 1.60e-02}$	$0.349_{\pm 1.55e-02}$	$31.004_{\pm 5.16e+00}$
DAOM-QC10	$0.348_{\pm 1.67e-02}$	$0.343_{\pm 1.68e-02}$	$4.843_{\pm 9.24e-01}$
DAOM-QC25	$0.354_{\pm 1.70e-02}$	$0.342_{\pm 1.69e-02}$	$5.583_{\pm 1.05e+00}$
PBGA 0.8 0.2	$0.313_{\pm 1.48e-02}$	$0.304_{\pm 1.39e-02}$	$15.620_{\pm 4.84e+00}$
PBGA 0.9 0.1	$0.321_{\pm 1.30e-02}$	$0.321_{\pm 1.28e-02}$	$2.024_{\pm 1.66e+00}$

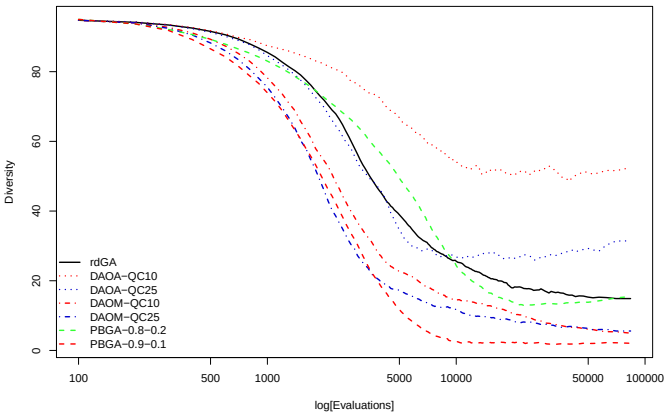
Table B.88: DAO vs. PBGA Convergence statistics for NK1-20-67 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.22: Convergence analysis DAO vs. PBGA *NK1-20-67*

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	0.00709 -	0.0337▲	0.0948▲	0.0831▲	0.119▲	0.118▲
DAOA-QC10		⁄	0.0266▲	0.0877▲	0.076▲	0.112▲	0.111▲
DAOA-QC25			⁄	0.061▲	0.0494▲	0.0857▲	0.0842▲
DAOM-QC10				⁄	−0.0117▽	0.0247▲	0.0231▲
DAOM-QC25					⁄	0.0363▲	0.0348▲
PBGA 0.8 0.2						⁄	−0.00153 -
PBGA 0.9 0.1							⁄

Table B.89: DAO vs. PBGA Convergence statistics for NK2-20-67 Cross comparison Average fitness

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	−0.0175▽	0.00665 -	0.0159▲	0.0098▲	0.0435▲	0.0323▲
DAOA-QC10		⁄	0.0242▲	0.0335▲	0.0273▲	0.061▲	0.0499▲
DAOA-QC25			⁄	0.0093▲	0.00316 -	0.0369▲	0.0257▲
DAOM-QC10				⁄	−0.00614 -	0.0276▲	0.0164▲
DAOM-QC25					⁄	0.0337▲	0.0225▲
PBGA 0.8 0.2						⁄	−0.0112▽
PBGA 0.9 0.1							⁄

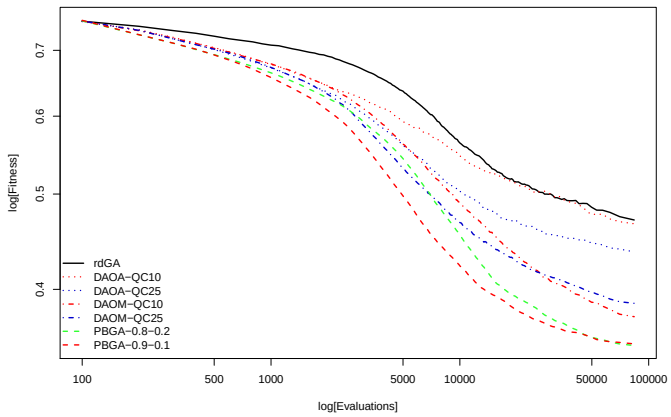
Table B.90: DAO vs. PBGA Convergence statistics for NK2-20-67 Cross comparison Best fitness

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	−37.853▲	−13.844▲	9.602▽	9.319▽	−1.023 -	13.011▽
DAOA-QC10		⁄	24.008▽	47.454▽	47.172▽	36.830▽	50.863▽
DAOA-QC25			⁄	23.446▽	23.163▽	12.821▽	26.855▽
DAOM-QC10				⁄	−0.283▲	−10.625▲	3.409▽
DAOM-QC25					⁄	−10.342▲	3.692▽
PBGA 0.8 0.2						⁄	14.034▽
PBGA 0.9 0.1							⁄

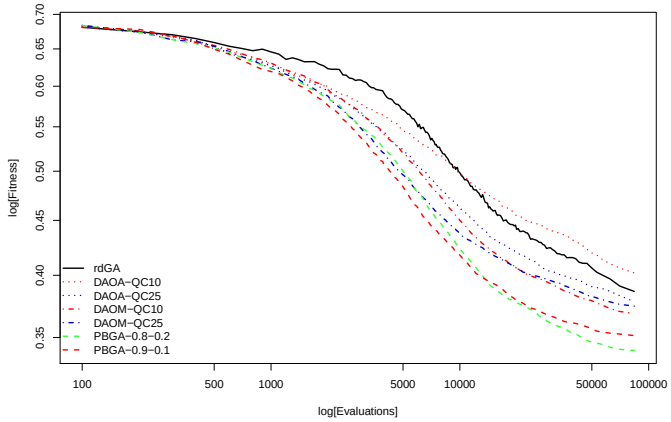
Table B.91: DAO vs. PBGA Convergence statistics for NK2-20-67 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.470_{\pm 1.37e-02}$	$0.384_{\pm 1.98e-02}$	$14.748_{\pm 1.63e+00}$
DAOA-QC10	$0.463_{\pm 2.38e-02}$	$0.401_{\pm 1.75e-02}$	$52.601_{\pm 1.01e+01}$
DAOA-QC25	$0.436_{\pm 1.45e-02}$	$0.377_{\pm 1.66e-02}$	$28.592_{\pm 4.62e+00}$
DAOM-QC10	$0.375_{\pm 1.57e-02}$	$0.368_{\pm 1.39e-02}$	$5.146_{\pm 1.36e+00}$
DAOM-QC25	$0.387_{\pm 1.51e-02}$	$0.374_{\pm 1.55e-02}$	$5.429_{\pm 8.93e-01}$
PBGA 0.8 0.2	$0.350_{\pm 1.74e-02}$	$0.340_{\pm 1.28e-02}$	$15.771_{\pm 8.51e+00}$
PBGA 0.9 0.1	$0.352_{\pm 1.36e-02}$	$0.351_{\pm 1.35e-02}$	$1.737_{\pm 1.39e+00}$

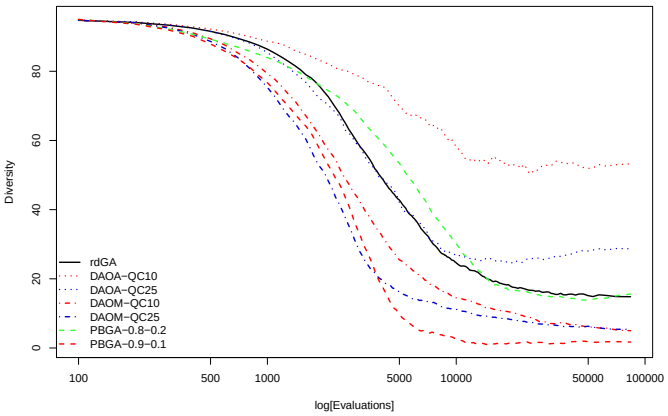
Table B.92: DAO vs. PBGA Convergence statistics for NK2-20-67 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.23: Convergence analysis DAO vs. PBGA *NK2-20-67*

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	−0.00272 -	0.0371▲	0.0467▲	0.0499▲	0.091▲	0.0773▲
DAOA-QC10		⁄	0.0398▲	0.0494▲	0.0526▲	0.0937▲	0.080▲
DAOA-QC25			⁄	0.00958 -	0.0128 -	0.0539▲	0.0402▲
DAOM-QC10				⁄	0.00324 -	0.0444▲	0.0306▲
DAOM-QC25					⁄	0.0411▲	0.0274▲
PBGA 0.8 0.2						⁄	−0.0137 -
PBGA 0.9 0.1							⁄

Table B.93: DAO vs. PBGA Convergence statistics for NK1-4-67 Cross comparison Average fitness

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	−0.0145 -	0.0102 -	−0.0149 -	−0.00676 -	0.0341▲	0.0129 -
DAOA-QC10		⁄	0.0247▲	−0.00046 -	0.00772 -	0.0486▲	0.0274▲
DAOA-QC25			⁄	−0.0251▽	−0.0169▽	0.0239▲	0.00271 -
DAOM-QC10				⁄	0.00818 -	0.049▲	0.0278▲
DAOM-QC25					⁄	0.0409▲	0.0197▲
PBGA 0.8 0.2						⁄	−0.0212▽
PBGA 0.9 0.1							⁄

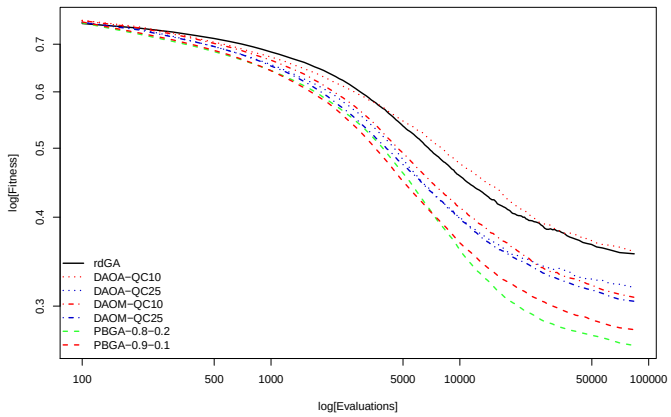
Table B.94: DAO vs. PBGA Convergence statistics for NK1-4-67 Cross comparison Best fitness

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	−39.409▲	−12.474▲	10.971▽	10.614▽	−12.689▲	8.723▽
DAOA-QC10		⁄	26.935▽	50.380▽	50.022▽	26.719▽	48.132▽
DAOA-QC25			⁄	23.445▽	23.087▽	−0.216 -	21.197▽
DAOM-QC10				⁄	−0.358 -	−23.661▲	−2.248▲
DAOM-QC25					⁄	−23.303▲	−1.890▲
PBGA 0.8 0.2						⁄	21.413▽
PBGA 0.9 0.1							⁄

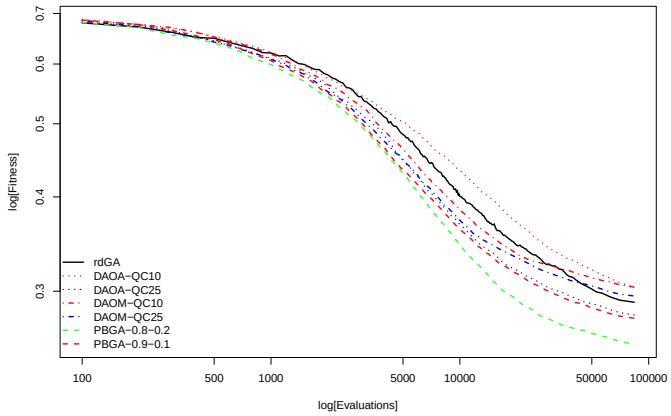
Table B.95: DAO vs. PBGA Convergence statistics for NK1-4-67 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.355_{\pm 3.82e-02}$	$0.288_{\pm 3.98e-02}$	$16.650_{\pm 1.49e+00}$
DAOA-QC10	$0.357_{\pm 3.84e-02}$	$0.303_{\pm 3.85e-02}$	$56.058_{\pm 6.86e+00}$
DAOA-QC25	$0.317_{\pm 3.85e-02}$	$0.278_{\pm 3.82e-02}$	$29.123_{\pm 5.07e+00}$
DAOM-QC10	$0.308_{\pm 4.22e-02}$	$0.303_{\pm 4.19e-02}$	$5.678_{\pm 8.80e-01}$
DAOM-QC25	$0.305_{\pm 3.37e-02}$	$0.295_{\pm 3.38e-02}$	$6.036_{\pm 8.37e-01}$
PBGA 0.8 0.2	$0.263_{\pm 3.78e-02}$	$0.254_{\pm 3.77e-02}$	$29.339_{\pm 4.84e+00}$
PBGA 0.9 0.1	$0.277_{\pm 3.25e-02}$	$0.276_{\pm 3.22e-02}$	$7.926_{\pm 3.34e+00}$

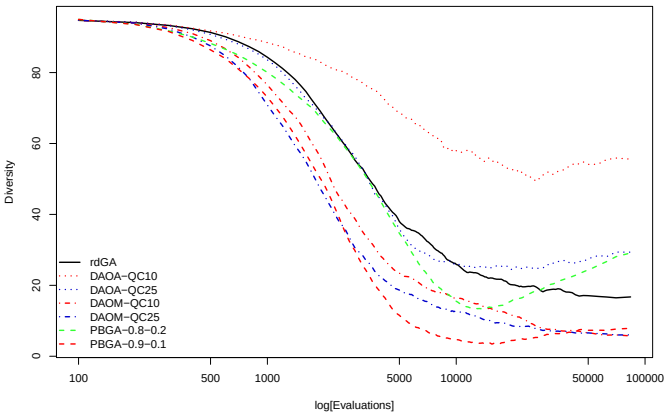
Table B.96: DAO vs. PBGA Convergence statistics for NK1-4-67 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.24: Convergence analysis DAO vs. PBGA *NK1-4-67*

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	−0.00521 -	0.0283▲	0.0604▲	0.0512▲	0.0907▲	0.0899▲
DAOA-QC10		⁄	0.0335▲	0.0657▲	0.0564▲	0.0959▲	0.0951▲
DAOA-QC25			⁄	0.0321▲	0.0229▲	0.0624▲	0.0616▲
DAOM-QC10				⁄	−0.00921▽	0.0303▲	0.0295▲
DAOM-QC25					⁄	0.0395▲	0.0387▲
PBGA 0.8 0.2						⁄	−0.000808 -
PBGA 0.9 0.1							⁄

Table B.97: DAO vs. PBGA Convergence statistics for NK1-20-97 Cross comparison Average fitness

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	−0.0126▽	0.014▲	0.0147▲	0.00885▲	0.0477▲	0.0391▲
DAOA-QC10		⁄	0.0266▲	0.0273▲	0.0215▲	0.0603▲	0.0517▲
DAOA-QC25			⁄	0.000699 -	−0.00514 -	0.0337▲	0.0251▲
DAOM-QC10				⁄	−0.00584 -	0.033▲	0.0244▲
DAOM-QC25					⁄	0.0388▲	0.0302▲
PBGA 0.8 0.2						⁄	−0.0086▽
PBGA 0.9 0.1							⁄

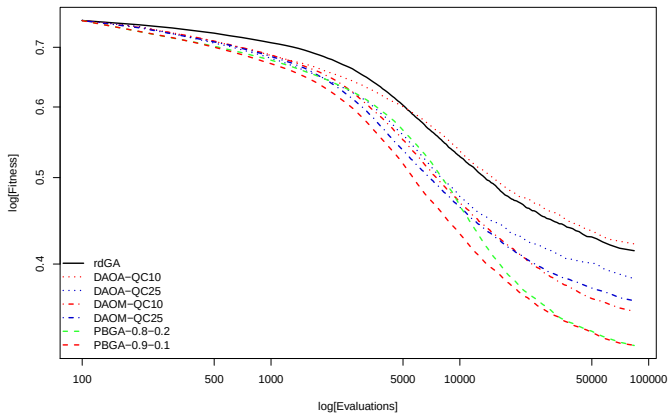
Table B.98: DAO vs. PBGA Convergence statistics for NK1-20-97 Cross comparison Best fitness

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	−29.333▲	−9.563▲	6.659▽	6.567▽	−3.633▲	8.404▽
DAOA-QC10		⁄	19.770▽	35.992▽	35.900▽	25.700▽	37.736▽
DAOA-QC25			⁄	16.222▽	16.130▽	5.930▽	17.967▽
DAOM-QC10				⁄	−0.0922 -	−10.292▲	1.744▽
DAOM-QC25					⁄	−10.200▲	1.837▽
PBGA 0.8 0.2						⁄	12.036▽
PBGA 0.9 0.1							⁄

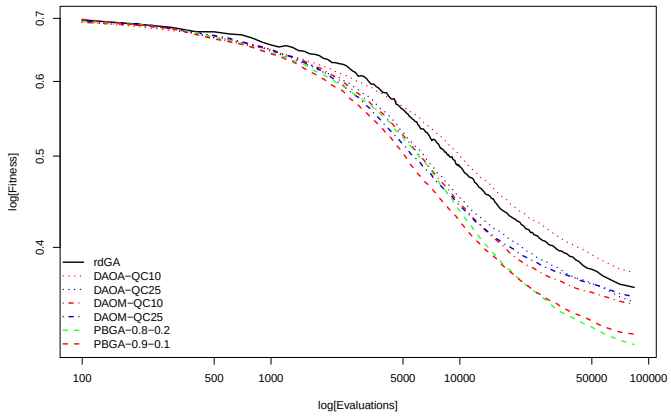
Table B.99: DAO vs. PBGA Convergence statistics for NK1-20-97 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.414_{\pm 1.10e-02}$	$0.362_{\pm 1.31e-02}$	$10.703_{\pm 1.28e+00}$
DAOA-QC10	$0.419_{\pm 1.49e-02}$	$0.375_{\pm 1.38e-02}$	$40.035_{\pm 6.33e+00}$
DAOA-QC25	$0.386_{\pm 1.54e-02}$	$0.348_{\pm 1.51e-02}$	$20.266_{\pm 3.34e+00}$
DAOM-QC10	$0.353_{\pm 1.31e-02}$	$0.348_{\pm 1.34e-02}$	$4.043_{\pm 9.29e-01}$
DAOM-QC25	$0.363_{\pm 1.17e-02}$	$0.354_{\pm 1.18e-02}$	$4.136_{\pm 8.61e-01}$
PBGA 0.8 0.2	$0.323_{\pm 1.20e-02}$	$0.315_{\pm 1.14e-02}$	$14.335_{\pm 3.82e+00}$
PBGA 0.9 0.1	$0.324_{\pm 1.33e-02}$	$0.323_{\pm 1.32e-02}$	$2.299_{\pm 1.61e+00}$

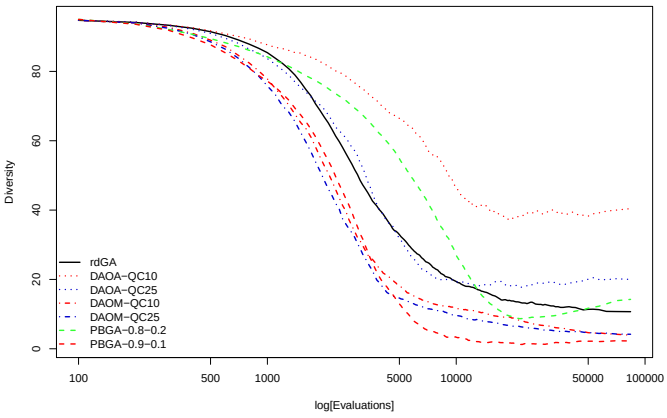
Table B.100: DAO vs. PBGA Convergence statistics for NK1-20-97 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.25: Convergence analysis DAO vs. PBGA *NK1-20-97*

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	−0.00685 -	0.0226▲	0.0677▲	0.0624▲	0.0987▲	0.0935▲
DAOA-QC10		⁄	0.0295▲	0.0745▲	0.0693▲	0.106▲	0.100▲
DAOA-QC25			⁄	0.0451▲	0.0398▲	0.0761▲	0.0709▲
DAOM-QC10				⁄	−0.00525 -	0.031▲	0.0258▲
DAOM-QC25					⁄	0.0363▲	0.0311▲
PBGA 0.8 0.2						⁄	−0.00518 -
PBGA 0.9 0.1							⁄

Table B.101: DAO vs. PBGA Convergence statistics for NK2-20-97 Cross comparison Average fitness

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	−0.0133▽	0.00268 -	0.0134▲	0.0114▲	0.0459▲	0.0338▲
DAOA-QC10		⁄	0.016▲	0.0267▲	0.0247▲	0.0592▲	0.0471▲
DAOA-QC25			⁄	0.0107▲	0.00875 -	0.0432▲	0.0311▲
DAOM-QC10				⁄	−0.00198 -	0.0325▲	0.0204▲
DAOM-QC25					⁄	0.0345▲	0.0223▲
PBGA 0.8 0.2						⁄	−0.0121▽
PBGA 0.9 0.1							⁄

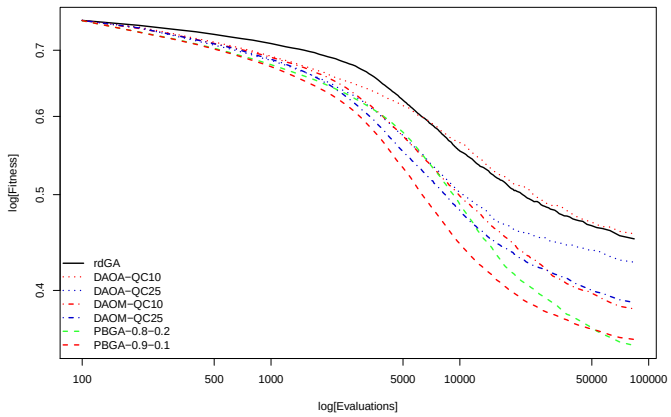
Table B.102: DAO vs. PBGA Convergence statistics for NK2-20-97 Cross comparison Best fitness

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	−28.110▲	−10.432▲	6.764▽	6.653▽	−0.650 -	8.795▽
DAOA-QC10		⁄	17.678▽	34.874▽	34.763▽	27.460▽	36.905▽
DAOA-QC25			⁄	17.196▽	17.085▽	9.782▽	19.227▽
DAOM-QC10				⁄	−0.110 -	−7.414▲	2.031▽
DAOM-QC25					⁄	−7.303▲	2.141▽
PBGA 0.8 0.2						⁄	9.444▽
PBGA 0.9 0.1							⁄

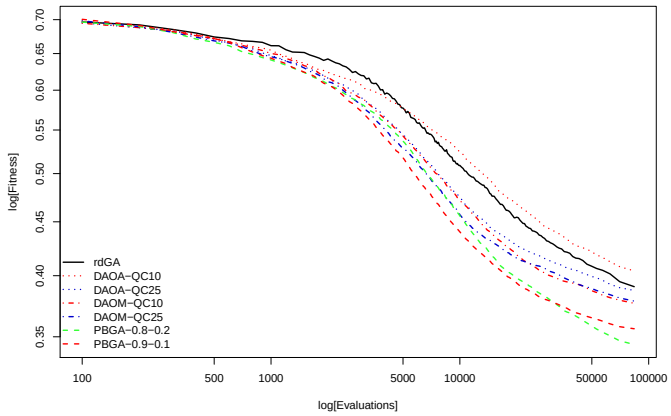
Table B.103: DAO vs. PBGA Convergence statistics for NK2-20-97 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.450_{\pm 1.03e-02}$	$0.389_{\pm 1.26e-02}$	$10.520_{\pm 1.04e+00}$
DAOA-QC10	$0.457_{\pm 1.84e-02}$	$0.403_{\pm 1.63e-02}$	$38.630_{\pm 5.97e+00}$
DAOA-QC25	$0.427_{\pm 1.57e-02}$	$0.387_{\pm 1.46e-02}$	$20.952_{\pm 4.44e+00}$
DAOM-QC10	$0.382_{\pm 1.52e-02}$	$0.376_{\pm 1.48e-02}$	$3.756_{\pm 8.72e-01}$
DAOM-QC25	$0.387_{\pm 1.47e-02}$	$0.378_{\pm 1.42e-02}$	$3.867_{\pm 7.60e-01}$
PBGA 0.8 0.2	$0.351_{\pm 1.44e-02}$	$0.343_{\pm 1.24e-02}$	$11.170_{\pm 4.04e+00}$
PBGA 0.9 0.1	$0.356_{\pm 1.13e-02}$	$0.355_{\pm 1.14e-02}$	$1.725_{\pm 1.39e+00}$

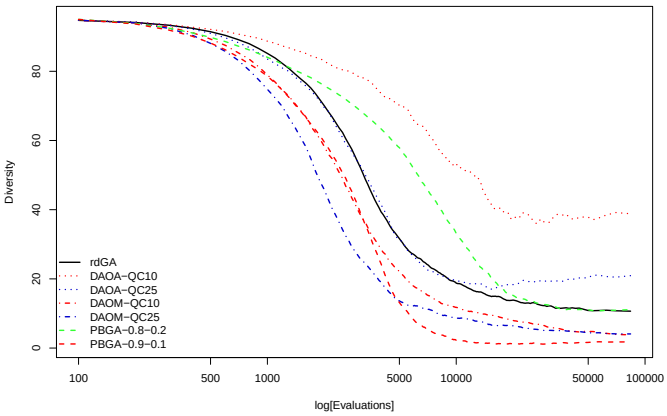
Table B.104: DAO vs. PBGA Convergence statistics for NK2-20-97 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.26: Convergence analysis DAO vs. PBGA *NK2-20-97*

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	0.0148 -	0.0184▲	0.0392▲	0.0383▲	0.0755▲	0.0612▲
DAOA-QC10		⁄	0.00356 -	0.0244▲	0.0235▲	0.0607▲	0.0464▲
DAOA-QC25			⁄	0.0208▲	0.0199▲	0.0572▲	0.0428▲
DAOM-QC10				⁄	−0.000883 -	0.0363▲	0.022▲
DAOM-QC25					⁄	0.0372▲	0.0229▲
PBGA 0.8 0.2						⁄	−0.0144 -
PBGA 0.9 0.1							⁄

Table B.105: DAO vs. PBGA Convergence statistics for NK1-4-97 Cross comparison Average fitness

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	0.00527 -	0.000733 -	−0.00176 -	−0.000155 -	0.040▲	0.0176▲
DAOA-QC10		⁄	−0.00453 -	−0.00702 -	−0.00542 -	0.0347▲	0.0123 -
DAOA-QC25			⁄	−0.00249 -	−0.000889 -	0.0393▲	0.0168 -
DAOM-QC10				⁄	0.0016 -	0.0418▲	0.0193▲
DAOM-QC25					⁄	0.0402▲	0.0177 -
PBGA 0.8 0.2						⁄	−0.0224▽
PBGA 0.9 0.1							⁄

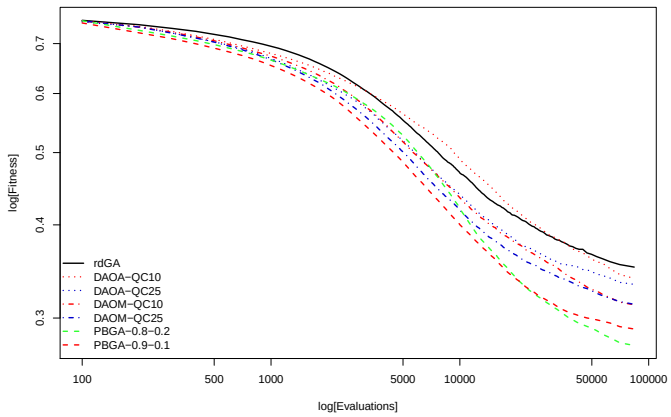
Table B.106: DAO vs. PBGA Convergence statistics for NK1-4-97 Cross comparison Best fitness

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	−29.413▲	−8.922▲	8.023▽	8.175▽	−12.146▲	5.150▽
DAOA-QC10		⁄	20.491▽	37.436▽	37.588▽	17.267▽	34.563▽
DAOA-QC25			⁄	16.944▽	17.096▽	−3.224▲	14.072▽
DAOM-QC10				⁄	0.152 -	−20.168▲	−2.872▲
DAOM-QC25					⁄	−20.320▲	−3.024▲
PBGA 0.8 0.2						⁄	17.296▽
PBGA 0.9 0.1							⁄

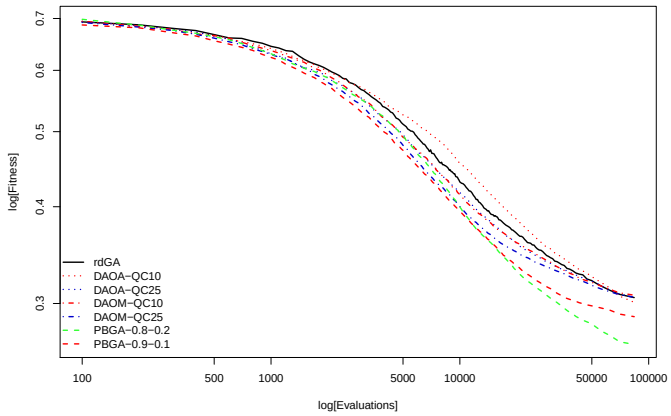
Table B.107: DAO vs. PBGA Convergence statistics for NK1-4-97 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.351_{\pm 2.50e-02}$	$0.305_{\pm 2.53e-02}$	$12.376_{\pm 1.06e+00}$
DAOA-QC10	$0.336_{\pm 3.70e-02}$	$0.300_{\pm 3.79e-02}$	$41.789_{\pm 4.90e+00}$
DAOA-QC25	$0.332_{\pm 3.11e-02}$	$0.304_{\pm 3.10e-02}$	$21.298_{\pm 3.47e+00}$
DAOM-QC10	$0.311_{\pm 2.60e-02}$	$0.307_{\pm 2.68e-02}$	$4.354_{\pm 7.47e-01}$
DAOM-QC25	$0.312_{\pm 3.89e-02}$	$0.305_{\pm 3.98e-02}$	$4.202_{\pm 6.49e-01}$
PBGA 0.8 0.2	$0.275_{\pm 3.00e-02}$	$0.265_{\pm 2.88e-02}$	$24.522_{\pm 3.90e+00}$
PBGA 0.9 0.1	$0.289_{\pm 3.24e-02}$	$0.288_{\pm 3.23e-02}$	$7.226_{\pm 2.52e+00}$

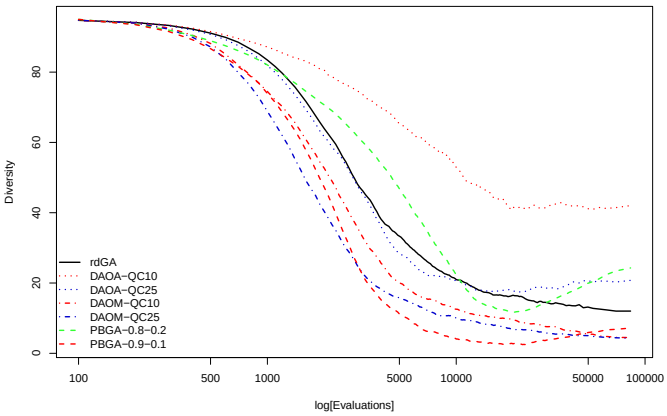
Table B.108: DAO vs. PBGA Convergence statistics for NK1-4-97 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.27: Convergence analysis DAO vs. PBGA *NK1-4-97*

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	0.000374 -	0.0294▲	0.0667▲	0.0491▲	0.0917▲	0.0811▲
DAOA-QC10		⁄	0.029▲	0.0664▲	0.0487▲	0.0914▲	0.0808▲
DAOA-QC25			⁄	0.0373▲	0.0197▲	0.0623▲	0.0517▲
DAOM-QC10				⁄	−0.0176▽	0.025▲	0.0144▲
DAOM-QC25					⁄	0.0426▲	0.032▲
PBGA 0.8 0.2						⁄	−0.0106▽
PBGA 0.9 0.1							⁄

Table B.109: DAO vs. PBGA Convergence statistics for NK2-4-97 Cross comparison Average fitness

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	−0.00789 -	0.0155▲	0.0206▲	0.00559 -	0.0483▲	0.0303▲
DAOA-QC10		⁄	0.0233▲	0.0285▲	0.0135▲	0.0562▲	0.0382▲
DAOA-QC25			⁄	0.00518 -	−0.00986▽	0.0329▲	0.0149▲
DAOM-QC10				⁄	−0.015▽	0.0277▲	0.00971 -
DAOM-QC25					⁄	0.0427▲	0.0247▲
PBGA 0.8 0.2						⁄	−0.018▽
PBGA 0.9 0.1							⁄

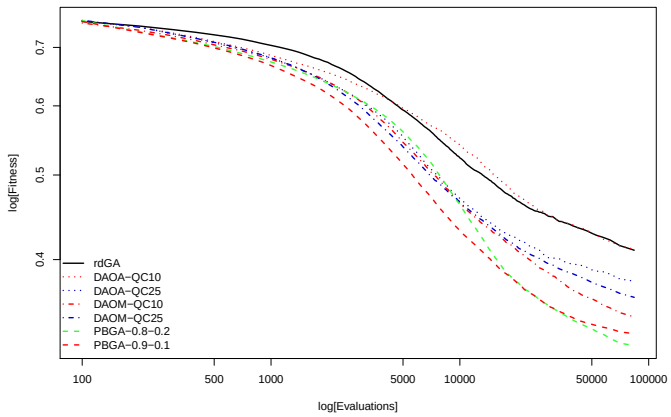
Table B.110: DAO vs. PBGA Convergence statistics for NK2-4-97 Cross comparison Best fitness

	rdGA	DAOA-QC10	DAOA-QC25	DAOM-QC10	DAOM-QC25	PBGA 0.8 0.2	PBGA 0.9 0.1
rdGA	⁄	−28.592▲	−11.266▲	7.247▽	7.059▽	−6.687▲	6.423▽
DAOA-QC10		⁄	17.326▽	35.839▽	35.651▽	21.905▽	35.015▽
DAOA-QC25			⁄	18.513▽	18.325▽	4.579▽	17.689▽
DAOM-QC10				⁄	−0.189 -	−13.934▲	−0.824 -
DAOM-QC25					⁄	−13.745▲	−0.636 -
PBGA 0.8 0.2						⁄	13.109▽
PBGA 0.9 0.1							⁄

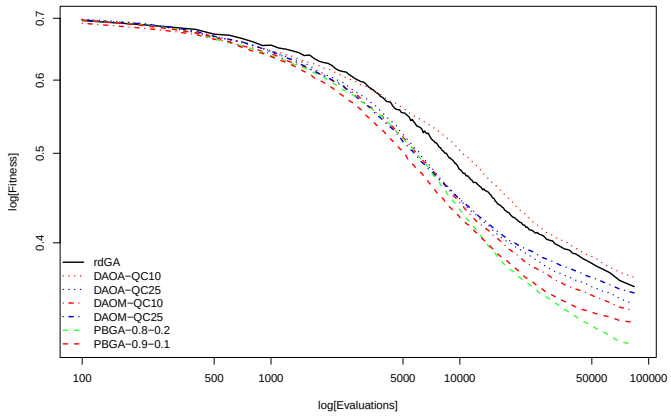
Table B.111: DAO vs. PBGA Convergence statistics for NK2-4-97 Cross comparison Average diversity

	Average fitness	Best fitness	Average diversity
rdGA	$0.410_{\pm 1.61e-02}$	$0.358_{\pm 1.79e-02}$	$11.422_{\pm 8.10e-01}$
DAOA-QC10	$0.409_{\pm 2.32e-02}$	$0.366_{\pm 2.27e-02}$	$40.014_{\pm 6.61e+00}$
DAOA-QC25	$0.380_{\pm 1.73e-02}$	$0.342_{\pm 1.90e-02}$	$22.688_{\pm 3.17e+00}$
DAOM-QC10	$0.343_{\pm 2.37e-02}$	$0.337_{\pm 2.40e-02}$	$4.175_{\pm 8.54e-01}$
DAOM-QC25	$0.361_{\pm 1.50e-02}$	$0.352_{\pm 1.52e-02}$	$4.363_{\pm 6.59e-01}$
PBGA 0.8 0.2	$0.318_{\pm 2.05e-02}$	$0.309_{\pm 2.05e-02}$	$18.108_{\pm 3.65e+00}$
PBGA 0.9 0.1	$0.329_{\pm 1.77e-02}$	$0.327_{\pm 1.76e-02}$	$4.999_{\pm 2.10e+00}$

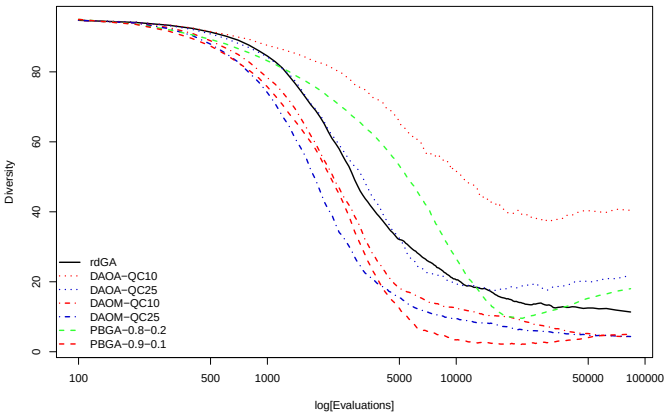
Table B.112: DAO vs. PBGA Convergence statistics for NK2-4-97 Final values Average diversity



(a) Average fitness



(b) Best fitness



(c) Average diversity

Figure B.28: Convergence analysis DAO vs. PBGA *NK2-4-97*